
DIPLOMARBEIT

Herr Ing.
Alexander Hanko

**Entwurf einer Datenbank-
lösung zur Verwaltung von
Transformatordaten und
zugehörigen Instand-
haltungsdaten**

Lahstedt, 2013

DIPLOMARBEIT

Entwurf einer Datenbank- lösung zur Verwaltung von Transformatorendaten und zugehörigen Instand- haltungsdaten

Autor:

Herr Ing. Alexander Hanko

Studiengang:

Informationstechnik

Seminargruppe:

KI09wF-D

Erstprüfer:

Prof. Dr.-Ing. Thomas Beierlein

Zweitprüfer:

Dr.-Ing. Alexander Horn

Einreichung:

Mittweida, 12.07.2013

Verteidigung/Bewertung:

Mittweida, 2013

Bibliografische Angaben:

Hanko, Alexander:

Entwurf einer Datenbanklösung zur Verwaltung von Transformatorendaten und zugehörigen Instandhaltungsdaten – 2013. – 89 Seiten.

Lahstedt, Hochschule Mittweida (FH), University of Applied Sciences,
Fakultät Elektro- und Informationstechnik, Diplomarbeit, 2013

Referat:

Die vorliegende Arbeit befasst sich mit der Entwicklung einer Datenbanklösung zur Verwaltung von Transformatorendaten. Einleitend wird eine Marktforschung durchgeführt, um eine passende Datenbankanwendung zu finden.

Der Hauptteil befasst sich mit dem Entwurf der Datenbanklösung. An dieser Stelle werden alle Möglichkeiten und Funktionen aufgeführt und erklärt.

Abschließend folgen die Integration des Datenbestandes in die Datenbank und die Funktionstests.

Inhaltsverzeichnis

| | |
|------------------------------------------------------|------------|
| Inhaltsverzeichnis | i |
| Abbildungsverzeichnis | iii |
| Tabellenverzeichnis | v |
| Abkürzungsverzeichnis | vi |
| 1 Übersicht | 1 |
| 1.1 Motivation | 1 |
| 1.2 Zielsetzung..... | 1 |
| 1.3 Salzgitter Flachstahl GmbH..... | 1 |
| 1.4 Kapitelgliederung | 3 |
| 2 Stand der Technik | 5 |
| 2.1 Analyse der Ist-Situation | 5 |
| 2.2 Verbesserung der Ist-Situation | 6 |
| 3 Präzisierung der Aufgabenstellung | 9 |
| 3.1 Zielparameter der Datenbank..... | 9 |
| 3.2 Darstellung und Nutzung der Datenbank..... | 10 |
| 4 Marktanalyse | 11 |
| 4.1 Marktrecherche | 11 |
| 4.1.1 SAP NetWeaver - ABAP | 12 |
| 4.1.2 Visual C# mit SQL-Server | 15 |
| 4.1.3 Microsoft Access | 18 |
| 4.2 Bewertung der Marktanalyse/Programmauswahl | 20 |
| 5 Systemkonzept..... | 23 |
| 5.1 Grundlagen der Datenbanken | 23 |
| 5.2 Datenbankmodelle | 23 |
| 5.2.1 Das hierarchische Datenbankmodell | 25 |
| 5.2.2 Das objektorientierte Datenbankmodell | 26 |
| 5.2.3 Das netzwerkförmige Datenbankmodell | 26 |

| | | |
|----------------------------------|-----------------------------------------------|-----------|
| 5.2.4 | Das relationale Datenbankmodell | 27 |
| 5.3 | Datenbankbegriffe..... | 28 |
| 5.4 | Beziehungen innerhalb einer Datenbank | 29 |
| 5.4.1 | 1-1 Beziehung..... | 30 |
| 5.4.2 | 1-c Beziehung..... | 31 |
| 5.4.3 | 1-m Beziehung..... | 31 |
| 5.4.4 | 1-mc Beziehung..... | 31 |
| 5.4.5 | ID-Schlüssel und Primär-/Fremdschlüssel | 31 |
| 5.5 | Normalisierungsprozess..... | 33 |
| 5.6 | Strukturregeln | 35 |
| 6 | Softwareentwurf..... | 37 |
| 6.1 | Strukturfestlegung..... | 37 |
| 6.2 | Tabellenzusammenstellung | 38 |
| 6.3 | Tabellenbeziehungen..... | 43 |
| 6.4 | Benutzermaske durch Formulare erstellen..... | 46 |
| 6.5 | Abfrage von Datenbeständen | 51 |
| 6.6 | Berichte aus Datenbeständen..... | 55 |
| 6.7 | Automatische Ölwertüberwachung..... | 58 |
| 7 | Softwareintegration und Test..... | 61 |
| 7.1 | Integration des Datenbestandes | 61 |
| 7.2 | Normalisierung der Datenbank | 62 |
| 7.3 | Test und Optimierung | 64 |
| 7.4 | Endkonzept der Datenbank..... | 64 |
| 8 | Zusammenfassung | 67 |
| 8.1 | Ergebnisse..... | 67 |
| 8.2 | Ausblick | 68 |
| Index | | 69 |
| Literatur | | 71 |
| Anhang | | I |
| Eidesstattliche Erklärung | | |

Abbildungsverzeichnis

| | |
|---------------------------------------------------------------------------------|----|
| Abbildung 1: Luftaufnahme der Salzgitter Flachstahl GmbH..... | 2 |
| Abbildung 2: Transformator 220/110kV | 3 |
| Abbildung 3: Aufgabenumfang der Datenbanklösung..... | 7 |
| Abbildung 4: Verlauf der Konzeptfestlegung..... | 8 |
| Abbildung 5: Selektionsbild aus dem SAP Beispielprogramm | 13 |
| Abbildung 6: Selektionsergebnisse aus dem SAP Beispielprogramm..... | 14 |
| Abbildung 7: Erstellung der Benutzeroberfläche unter C# | 16 |
| Abbildung 8: Quellcode für die Anbindung an eine Access-Datenbank | 17 |
| Abbildung 9: Tabellen einer Datenbank in Relation zueinander [H10 - S.122]..... | 18 |
| Abbildung 10: Übersicht der Access Programmoberfläche..... | 19 |
| Abbildung 11: Schalenmodell eines RDBMS nach [S09 - S. 8]..... | 24 |
| Abbildung 12: Aufbau einer hierarchischen Datenbank | 25 |
| Abbildung 13: Tabellenorganisation bei relationalen Datenbanken..... | 27 |
| Abbildung 14: Datenbankbegriffe | 28 |
| Abbildung 15: Entitätenblockdiagramm einer 1-1 Beziehung..... | 30 |
| Abbildung 16: Entitätenblockdiagramm einer 1-c Beziehung | 31 |
| Abbildung 17: Entitätenblockdiagramm einer 1-m Beziehung..... | 31 |
| Abbildung 18: Entitätenblockdiagramm einer 1-mc Beziehung | 31 |
| Abbildung 19: Primär- und Fremdschlüssel für Tabellenbeziehungen | 32 |
| Abbildung 20: Übersicht der Abhängigkeiten..... | 33 |

| | |
|-----------------------------------------------------------------------------|----|
| Abbildung 21: Ablaufdiagramm für den Entwurfsprozess | 37 |
| Abbildung 22: Menüleiste "Erstellen" in Access 2010 | 42 |
| Abbildung 23: Tabellenerstellung unter Access 2010 | 42 |
| Abbildung 24: Beziehungen zwischen den Tabellen | 44 |
| Abbildung 25: Beziehungseinstellungen unter Access 2010..... | 44 |
| Abbildung 26: Formular „Trafodaten“ in der Entwurfsansicht..... | 46 |
| Abbildung 27: Benutzeroberfläche des Formulars „Trafodaten“ | 48 |
| Abbildung 28: Unterformular „Maße“ zur Datenbankanwendung | 49 |
| Abbildung 29: Unterformular „Ölproben“ zur Datenbankanwendung | 49 |
| Abbildung 30: Unterformular „Standort/Kammern“ zur Datenbankanwendung | 50 |
| Abbildung 31: Unterformular „Stammdaten“ zur Datenbankanwendung..... | 51 |
| Abbildung 32: Entwurfsoberfläche für Abfragen in Access 2010 | 52 |
| Abbildung 33: Erstellung der Suchkriteriums-Felder unter Access 2010..... | 53 |
| Abbildung 34: Abfragefunktion in Access 2010 | 55 |
| Abbildung 35: Bericht zur Trafoauswertung in Access 2010..... | 56 |
| Abbildung 36: Unterformular „Auswertung“ zur Datenbankanwendung | 57 |
| Abbildung 37: Makrogenerator für die Daten-Konvertierung zu Excel | 58 |
| Abbildung 38: Tabelle „Grenzwerte“ für den Öldatenabgleich | 59 |
| Abbildung 39: Unterformular „Grenzwerte für Ölproben“ | 60 |
| Abbildung 40: Integration der Bestandsdaten..... | 61 |
| Abbildung 41: Benutzeroberfläche der Datenbankanwendung | 65 |

Tabellenverzeichnis

| | |
|------------------------------------------------------------------|----|
| Tabelle 1: Übersicht der Datenbankinhalte | 9 |
| Tabelle 2: Datenbankanwendungseigenschaften - must have | 21 |
| Tabelle 3: Datenbankanwendungseigenschaften - can have | 21 |
| Tabelle 4: Datenbankanwendungseigenschaften - nice to have | 22 |
| Tabelle 5: Assoziationstypen | 29 |
| Tabelle 6: Beziehungstypen | 30 |

Abkürzungsverzeichnis

| | |
|--------------|----------------------------------------------------------------|
| ABAP | Advanced Business Application Programming (Programmiersprache) |
| C# | C-Sharp |
| DB | Datenbank |
| DBMS | Datenbank Management System |
| EDV | Elektronische Datenverarbeitung |
| ER | Entity-Relationship |
| GUI | Graphical User Interface |
| ID | Identifikation |
| IT | Informationstechnik |
| PC | Personal-Computer |
| PCB | Polychlorierte Biphenyle |
| PDF | Portable Document Format |
| RDBMS | Relationales Datenbank Management System |
| SAP | Systemanalyse und Programmentwicklung (Softwarefirma) |
| SQL | Structured Query Language |
| UK | <i>Kurzschlussspannung</i> |
| VBA | Visual Basic for Applications |
| VDE | Verband der Elektrotechnik |
| WHG | Wasserhaushaltsgesetz |

1 Übersicht

1.1 Motivation

In der heutigen Zeit wird die tägliche Arbeit immer stärker durch EDV-Systeme¹ unterstützt. Es wird ein Wandel von einer papiergestützten Arbeit hin zur computergestützten Arbeit vollzogen. Gerade bei der Verwaltung von Gerätschaften und Bauteilen ist der Einsatz von Computersystemen unabdingbar geworden.

Im Arbeitsbereich der Energiebetriebe Salzgitter² gibt es eine Vielzahl von Transformatoren, die verwaltet werden müssen. Die Anforderung besteht darin, eine Datenbanklösung zu entwickeln, die dieser Aufgabe gerecht wird. Es müssen neben den Kenndaten der Transformatoren auch Daten zum Wasserhaushaltsgesetz und Instandhaltungsdaten gespeichert und verwaltet werden. Eine Verwaltung auf Papierbasis ist aufgrund des Umfangs nicht mehr praktikabel. Eine elektronische Lösung ermöglicht des Weiteren die Verknüpfung von technischen Daten. So ist deren Auswertung möglich und die Daten können für die vorbeugende Instandhaltung genutzt werden. Also um Fehler oder Defekte beispielsweise an Transformatoren frühzeitig zu erkennen.

1.2 Zielsetzung

Um den Anforderungen gerecht zu werden, wird eine Datenbankanwendung entwickelt, in der sich alle erforderlichen Daten speichern und verwalten lassen. Zum Finden einer geeigneten Datenbanktechnologie wird eine Marktforschung durchgeführt. Anschließend wird die Datenbank gegebenenfalls programmiert und mit Inhalten gefüllt. Möglichkeiten für die Datenauswertung und Ausgabefunktionen sollen ebenfalls vorhanden sein. Diese Funktionen sind für die Technik und den Umweltdienst bei den Energiebetrieben Salzgitter erforderlich. Eine Implementierungs- und Testphase werden den Rahmen der Arbeit komplettieren.

1.3 Salzgitter Flachstahl GmbH

Die Datenbank zur Transformatoren-Verwaltung wird nach ihrer Fertigstellung bei der Salzgitter Flachstahl GmbH eingesetzt. Bei ihr handelt es sich um ein Tochterunternehmen der Salzgitter AG. Die Salzgitter Flachstahl GmbH stellt Flachprodukte aus Stahl her. Diese werden an Fahrzeughersteller, Röhren- und Großröhrenhersteller sowie an Kalt-

¹ EDV-Systeme ist ein Sammelbegriff, welcher die Datenverarbeitung und Datenverarbeitungssysteme umfasst, die nicht mechanisch, sondern vor allem elektronisch arbeiten.

² Energiebetriebe Salzgitter ist eine Abteilung mit dem Aufgabengebiet der Stromversorgung innerhalb der Salzgitter Flachstahl GmbH – siehe auch Kapitel 1.3 Salzgitter Flachstahl GmbH

walzer³ und die Bauindustrie geliefert. Am Firmenstandort in Salzgitter sind circa 4.700 Mitarbeiter tätig, die jährlich etwa 4,5 Millionen Tonnen Rohstahl verarbeiten.



Abbildung 1: Luftaufnahme der Salzgitter Flachstahl GmbH

Bei der Salzgitter Flachstahl GmbH handelt es sich um ein integriertes Hüttenwerk. Dies bedeutet, dass es sich um eine Kombination aus mehreren Fertigungsstätten an einem Standort handelt, um an ihm aus Rohstoffen Stahlprodukte zu erzeugen. Am Standort Salzgitter gibt es in Produktionsreihenfolge aufgelistet die Betriebsteile Sinteranlage, Kokerei, Hochofen, Stahlwerk, Stranggussanlage, Walzwerk 1-4 und diverse Nebenanlagen.

Die erforderliche Energie für all diese Produktionsprozesse wird im werkeigenen Kraftwerk erzeugt. Das Gaskraftwerk wird mit Gasen aus den Hüttenprozessen geheizt, die sonst am Standort Salzgitter als „Abfall“ ungenutzt bleiben würden. Über ein 220/110kV Stromnetz wird die Energie in die Betriebsteile übertragen. Auf der 220kV Ebene existiert auch eine Anbindung des Standortes Salzgitter an das europäische Verbundnetz.

Um die Spannung in handelbare Größen zu wandeln, werden in den einzelnen Betriebsteilen Transformatoren benötigt (für ein Beispiel siehe Abbildung 2). Diese transformieren die Spannung von 220kV bis auf 400V herunter. Es gibt noch Zwischenebenen, je nachdem welche Spannung und auch Leistung jeweils benötigt wird. Die Spannungsebenen am Standort sind 220kV, 110kV, 30kV, 6kV, 690V, 500V, 400V. Insgesamt sind circa 520 Transformatoren eingesetzt.

³ Kaltwalzer – Firmen die Kaltwalzprodukte verarbeiten



Abbildung 2: Transformator 220/110kV

1.4 Kapitelgliederung

Die Diplomarbeit besteht aus acht Kapiteln. Nach der allgemeinen Einleitung des ersten Kapitels wird im **Kapitel 2** die Bestandssituation analysiert und ein Weg zur Optimierung aufgezeigt.

Anschließend wird im **Kapitel 3** die Aufgabenstellung weiter präzisiert. Dafür werden die Anforderungen, die durch das themenstellende Unternehmen vorgegeben werden, aufgezeigt und näher erläutert. Diese Anforderungen werden mithilfe eines Pflichtenheftes festgehalten. Zusätzlich wird der spätere Nutzen dargestellt.

Darauf folgend wird im **Kapitel 4** eine Marktanalyse durchgeführt. Dabei werden drei Programmieroberflächen vorgestellt, mit deren Hilfe die Umsetzung der Datenbank verwirklicht werden kann. Anschließend werden die Vor- und Nachteile dieser Oberflächen gegenübergestellt. Die bestmögliche Lösung wird im Kapitel 6 beim Softwareentwurf genutzt.

Im **Kapitel 5** werden die Grundbegriffe zu der Datenbankanwendung erläutert. Diese werden nach der Wahl der Programmierumgebung im Kapitel 4 zielgerichtet dargestellt. Nach dem Kapitelstart durch die Grundbegriffserklärung folgen anschließend die Erläuterungen zu den Datenbankthematiken, wie beispielsweise die Beziehungen zwischen den Tabellen einer Datenbank.

Im **Kapitel 6** erfolgt die Umsetzung der Anforderungen. Dafür wird zunächst die Struktur der Datenbank festgelegt. Im Anschluss folgen die Tabellenzusammenstellung und das Erstellen der Benutzermaske. Abschließend wird im Kapitel 6 eine Suchfunktion und eine Berichtserstellung implementiert. Auch eine automatische Überwachung von Dateninhalten wird kurz erläutert. Diese wird in Teilen durch Visual Basic⁴ (VBA) realisiert.

Im **Kapitel 7** schließt die Softwareintegration an, der vorhandene Datenbestand wird integriert. Bislang händisch geführte Daten werden ebenfalls in die Datenbank übernommen. Innerhalb der Datenbankanwendung folgen Kontrollen auf doppelt geführte Daten. Abschließende Tests zur Funktion der Datenbank werden mit unternehmensgebundenen Daten durchgeführt. Dabei auftretende Fehler werden aufgezeigt und die Behebung erklärt. Kapitelabschließend wird beschrieben, wie die funktionsfähige Datenbank beim themengebenden Unternehmen in Dienst gestellt wird.

Schließlich werden im **Kapitel 8** die Resultate der einzelnen Kapitel der Diplomarbeit zusammengefasst. Hier werden auch die Leistungen des Diplomanden aus seiner Sicht skizziert. Zusätzlich wird ein Ausblick auf mögliche Weiterentwicklungen gegeben.

⁴ Visual Basic for Applications (VBA) ist eine makrobasierte Skriptsprache von und für Microsoft Produkte. Sie dient der Steuerung von Programmabläufen.

2 Stand der Technik

2.1 Analyse der Ist-Situation

Aktuell wird beim themengebenden Unternehmen, der Salzgitter Flachstahl GmbH, eine mehrgleisige Verwaltung von Transformatoren geführt. Diese findet teilweise in EDV-gestützter-Form und teilweise in Papierform statt. In einer vorhandenen Access-Datenbank⁵ sind bislang Nerndaten zu den Transformatoren gespeichert. Sie sollen in einer zukünftigen Datenbanklösung wieder verwendet werden. Eine Übersicht dieser Daten ist in der nachfolgenden Auflistung zu finden:

- Hersteller
- Trafo-Typ
- Trafo-Fabriknummer
- Baujahr
- Abmessungen des Transformators
- Stand der letzten Öl-Probe
- Primärspannung
- Sekundärspannung
- Primärstrom
- Sekundärstrom
- Schaltgruppe
- Kurzschlussspannung
- Standort des Transformators

Für den Betrieb und die Zustandsbewertung eines Transformators ist es auch wichtig, die Werte des Isolieröls zu prüfen. Dieses dient der Isolation zwischen den elektrischen Wicklungen. Die Ergebnisse dieser Prüfungen werden in einer Akte abgelegt und archiviert. Eine einfache Übersicht der Wartungsdaten ist somit nicht gegeben. Die Einhaltung der Wartungsintervalle wird händisch, durch den jeweiligen Fachbetrieb geprüft.

Einen dritten Schwerpunkt zu den Transformatoren stellen die Umspannkammern dar. In ihnen sind die Transformatoren örtlich installiert. Es handelt sich somit um den Standort des Transformators. Bei der Wahl eines Standortes ist es wichtig, dass die Trafokammer in ihrer Statik nicht durch einen zu großen Umspanner überlastet wird. Daher sind die Gewichtsangaben und Abmessungen wichtig. Genauso müssen auch die Spurmaße⁶ bei der Standortwahl mit beachtet werden. Nicht alle Trafokammern sind für die gleichen Rollenabstände ausgelegt.

⁵ Access ist eine Programmierumgebung für Datenbankanwendungen von der Firma Microsoft.

⁶ Das Spurmaß gibt den Abstand zwischen den Fahrrollen eines Transformators an.

Wie bereits erwähnt, sind die Transformatoren mit Isolieröl gefüllt. Dieses stellt eine potentielle Gefahr für die Umwelt dar. Im Störfall könnte Öl aus den Transformatoren austreten und ins Erdreich gelangen. Damit dieses nicht geschehen kann, sind die Trafokammern mit einer Auffangwanne versehen, welche das Öl im Schadensfall aufnehmen kann. Da jede Auffangwanne nur ein begrenztes Volumen hat, darf nicht jeder Transformator auf jede Auffangwanne gestellt werden. Es müssen also bei einem Austausch oder einer Erneuerung eines Transformators auch die Örtlichkeiten bedacht werden. Alle Daten zu den Auffangwannen müssen durch den Umweltdienst abgenommen und regelmäßig wiederholend überprüft werden, da sonst die Auflagen des Wasserhaushaltsgesetzes⁷ (WHG) nicht erfüllt sind. Gemäß dieser Gesetzgebung muss jede Veränderung des Transformatorstandortes bekannt gegeben werden, vorausgesetzt, der Transformator hat ein bestimmtes Volumen und ist damit anzeigepflichtig. Alle diese Daten sind ebenfalls in Aktenform nachgehalten.

2.2 Verbesserung der Ist-Situation

Aufgrund des immer umfangreicher werdenden Datenbestandes und einer Zunahme an Papierverwaltung wird der Wunsch nach einer Datenbanklösung für alle Funktionen geäußert.

Wie in der Abbildung 3 zu sehen ist, sollen die vier Hauptbereiche in einer Datenbank zusammengefasst werden. So gibt es nur noch eine Verwaltungsinstanz, in der alle Daten zentral gespeichert sind. Mit einer solchen Datenbanklösung könnten auch die Instandhaltungsdaten automatisiert an die Fachbetriebe gesendet werden. Dieses ist beispielsweise die Elektro-Instandhaltung der Energiebetriebe Salzgitter.

Um die Ist-Situation zu verbessern, wurde in Besprechungen mit den betroffenen Abteilungen der Datenbankumfang festgelegt. Dazu erfolgte die Nennung der Daten von jedem Fachbereich. Bei der Konzeptfestlegung wird besonderes Augenmerk auf folgende Punkte gelegt:

- Wunsch einer einfachen Datenverwaltung
- Einfache Übersicht der auffälligen Umspanner
- Schnelle und einfache Suchfunktionen
- Leichtere Datenzusammenfassung und Ausgabe zur Weiterleitung an den Vorgesetzten beziehungsweise die Stabsabteilung

⁷ Das Wasserhaushaltsgesetz bildet den Hauptteil des deutschen Wasserrechts. Das WHG enthält unter anderem Bestimmungen zum Schutz des Grundwassers.

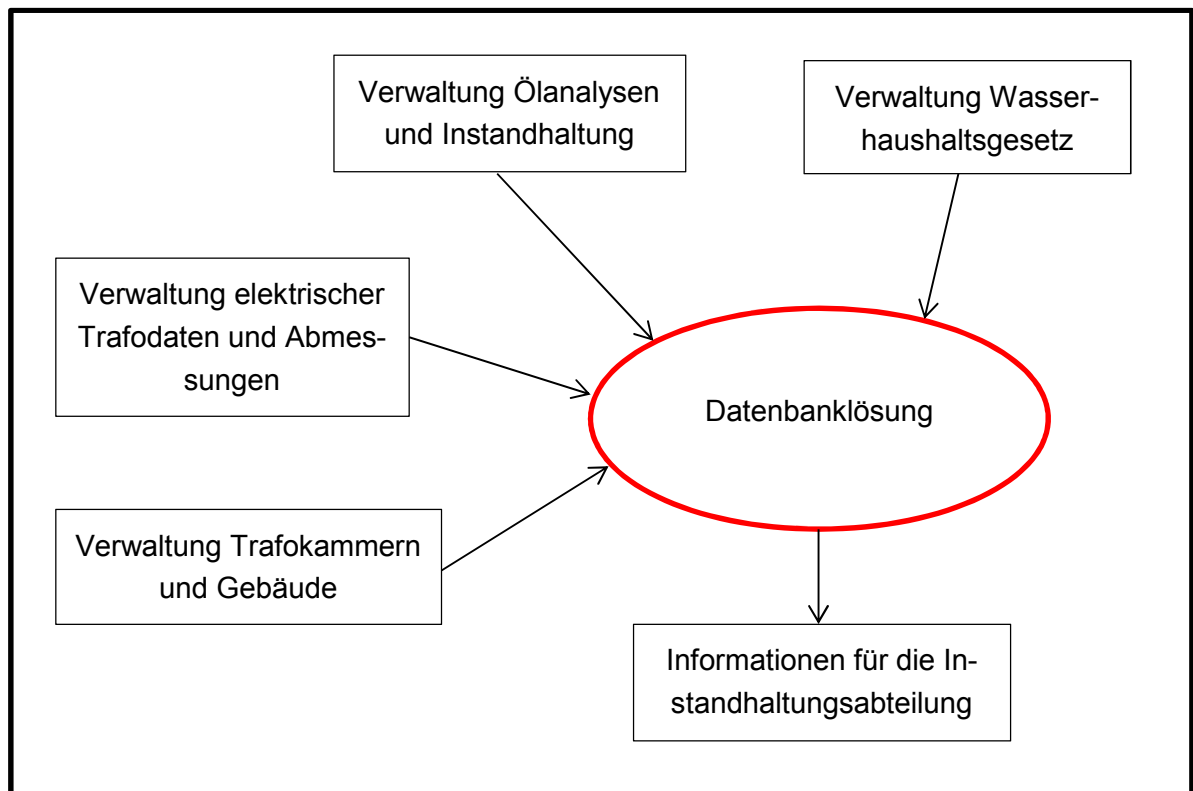


Abbildung 3: Aufgabenumfang der Datenbanklösung

Vorgegangen wird bei den Besprechungen in drei Schritten, welche in Abbildung 4 dargestellt sind. Nach einer ersten Vorbesprechung in dem jeweiligen Bereich werden die Anforderungen abteilungsintern abgeklärt. Bei einer zweiten, individuellen Besprechung werden die benötigten Inhalte für die Datenbankanwendung an das Konzept angepasst. Nach weiteren abteilungsinternen Abstimmungen kann schlussendlich der Inhalt für die Datenbankanwendung festgelegt werden.

In den Bereichen der E-Technik und der Gebäudetechnik mussten hauptsächlich abteilungsinterne Inhalte für die Datenbankanwendung abgestimmt werden. In dem Bereich des Wasserhaushaltsgesetzes müssen auch Gesetzauflagen beachtet werden. Somit erfolgt in diesem Bereich eine zusätzliche Abstimmung mit dem Umweltschutz der Salzgit-ter Flachstahl GmbH. Dieses ist eine Stabsabteilung, die sich ausschließlich mit den Belangen des Schutzes für Natur und Umwelt beschäftigt.

Die Koordination zwischen den drei Bereichen oblag dem Autor dieser Diplomarbeit. Dabei musste besondere Acht auf Schnittstellen zwischen den Bereichen gegeben werden, also welche Inhalte auch für den jeweils anderen Bereich von Bedeutung sind. Die Darstellung der Ergebnisse dieser Besprechungen lassen sich im nachfolgenden Kapitel wiederfinden.

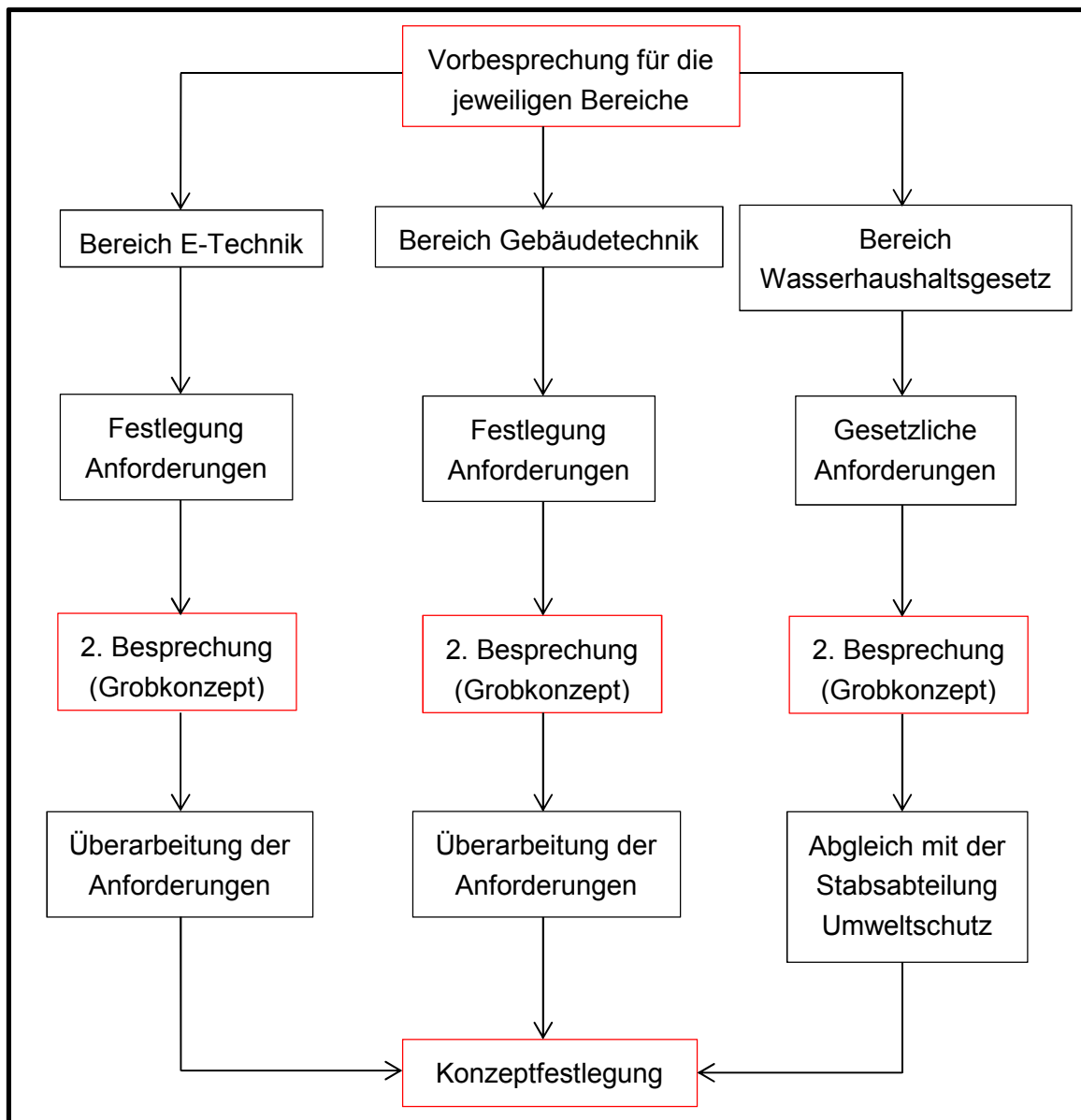


Abbildung 4: Verlauf der Konzeptfestlegung

3 Präzisierung der Aufgabenstellung

3.1 Zielparameter der Datenbank

Anhand der Besprechungsergebnisse aus Kapitel 2.2 werden die Parameter für die Datenbankanwendung festgelegt, welche von den betroffenen Abteilungen beigesteuert wurden. Zunächst geht es lediglich um den zu verwaltenden Inhalt, also die Datenbankinhalte. Aus diesen ergibt sich die nachfolgende Tabelle 1.

| Anforderungen E-Technik | Anforderungen Gebäude und WHG |
|---------------------------------------|--------------------------------------|
| Trafo-Hersteller | Standort |
| Fabriknummer zum Transformator | Fassungsvermögen der Auffangwanne |
| Typ | Belastungsvermögen der Trafokammer |
| Isolierung | Statik zu den Trafokammern vorhanden |
| Baujahr | Eigentümer der Trafokammer |
| Elektrische Leistungsdaten (komplett) | Eigentümer des Transformators |
| Abmessungen des Transformators | Verwaltungsnummer zur Auffangwanne |
| Verwaltung der Ölanalysen | Isolierölmenge (Erfassung für WHG) |

Tabelle 1: Übersicht der Datenbankinhalte

Wie in der vorhergehenden Tabelle 1 zu erkennen ist, gibt es Anforderungen, die das Wasserhaushaltsgesetz betreffen und Anforderungen von der E-Technik zu den Inhalten der Datenbankanwendung. Zu sehen sind Nenndaten eines Transformators zusammen mit Wartungsdaten, in der linken Spalte. In der rechten Spalte sind unter anderem Daten zu gesetzlichen Anforderungen zu finden. So muss beispielsweise das Fassungsvermögen der Auffangwanne zur Isolierölmenge des Transformators passen. Zusätzlich sind dort Daten zur Trafokammer zu finden. Diese sind erforderlich, um eine Überlastung einer Kammer zu vermeiden. Des Weiteren werden die Abmessungen einer Trafokammer benötigt, um bei einer Standortsuche passende Trafokammern zu finden. Verdeutlichung finden diese Daten noch einmal im Kapitel 6 bei der Zusammenstellung der Datenbankanwendung. Dort werden sie im Detail aufgeschlüsselt, um mit den Daten arbeiten zu können.

3.2 Darstellung und Nutzung der Datenbank

Neben den Inhalten der Datenbankanwendung wurde sich auch auf die Darstellung und Nutzung der Daten geeinigt. Im Folgenden wurde ein Pflichtenheft erstellt. In diesem sind neben den Inhalten der Datenbankanwendung die Anforderungen zu finden. Stichpunktartig seien folgende Eckpunkte des Pflichtenheftes genannt:

- Speicherung aller Transformatorenenddaten (siehe Tabelle 1)
- Speicherung von Ölanalysen
- Auswertung von Ölanalysen anhand von Schwellwerten
- Automatische Warnfunktion bei Überschreitung von Ölgrenzwerten
- Automatische Plausibilitätsprüfung bei Dateneingabe
- Suchfunktion für Transformatoren
- Speicherung von Trafokammerdaten
- Speicherung von Überprüfungsintervallen zu den Auffangwannen
- Zuordnung von Transformatoren zu den Trafokammern
- Pflege von Instandhaltungsdaten

Neben den aufgeführten Eckpunkten sollen zudem die folgenden funktionalen Eigenschaften, von der Datenbankanwendung erfüllt werden:

- Speicherung von externen Dokumenten in der Datenbank (ohne Revisionsstand)
- Export von Datensätzen zu Excel oder als Textdokument
- Druckausgabe von Suchergebnissen oder einzelnen Datensätzen
- Anordnung aller Elemente auf einer Benutzermaske

Die Wahl der Programmierumgebung wird dem Diplomanten überlassen. Die gestellten Anforderungen müssen jedoch erfüllbar bleiben. Auch eine Bedienung der späteren Datenbankanwendung für Computerlaien muss möglich sein. Zielsetzung ist eine möglichst kostenneutrale Umsetzung. Es sollen nach Möglichkeit keine externen Kosten für das themengebende Unternehmen anfallen. So müssen eventuell anfallende Lizenzkosten mit berücksichtigt werden. Nach einer durchzuführenden Marktanalyse soll das Ergebnis zunächst vorgestellt werden, bevor mit der Programmierung der Datenbankanwendung begonnen wird.

4 Marktanalyse

4.1 Marktrecherche

Für den Umfang der Marktforschung wurde seitens des themengebenden Unternehmens festgelegt, dass es sich bei der Datenbankanwendung um eine eigene Programmierung auf Basis bereits eingesetzter Programme handeln soll. Daher werden fertige Softwarelösungen, in die nur noch die Daten eingefügt werden müssten, bei der Marktrecherche ausgeschlossen. Diese Maßnahme dient der möglichen Kostenneutralität und zur Vereinfachung der Bedienung.

Einleitend werden zunächst die Kriterien für die Marktforschung vorgestellt. Weitere Voraussetzungen für die Programmierumgebung sind bereits im Kapitel 3 zu finden. Auf Grundlage dieser Vorgaben werden die Programme durchleuchtet. Die folgende Kriterien-sammlung für die Beurteilung von Datenbank-Programmiersprachen wird in [A91] eingeführt:

- **Abgeschlossenheit:** Das Ergebnis einer Abfrage ist wieder eine Relation⁸ und kann als Ergebnis für die nächste Abfrage verwendet werden.
- **Mengenorientiertheit:** Jede Operation soll auf Mengen von Daten gleichzeitig arbeiten, nicht navigierend auf einzelnen Elementen.
- **Effizienz:** Jede Operation ist effizient ausführbar.
- **Vollständigkeit:** Die Sprache muss mindestens die Anfrage einer Standardsprache⁹ ausdrücken können.

⁸ Eine Relation stellt im Bereich der relationalen Datenbanken eine Tabelle dar. Für nähere Erläuterungen siehe auch Kapitel 5.2.4.

⁹ Mit Standardsprache ist eine relationale Datenbanksprache gemeint, beispielsweise SQL (Structured Query Language) zur Definition der Datenstruktur sowie Abfrage und Manipulation der Daten.

Für die Beurteilung und Findung einer geeigneten Programmieranwendung werden des Weiteren noch folgende Kriterien mit untersucht:

- Wo werden die Daten gespeichert (zentral auf einem Server oder lokal auf einem Desktop PC)?
- Erfolgt die Programmierung und Betreuung der Datenbank im Unternehmen zentral oder durch die anwendende Abteilung?
- Wie viele Nutzer können gleichzeitig mit der Datenbank arbeiten?
- Kann die Datenbank auf mobilen Geräten genutzt werden?
- Können Konvertierungen zu anderen Programmen integriert werden (Microsoft Word, Excel, Outlook)?
- Kann die Datenbank mit anderen Programmen verknüpft werden oder lediglich als „stand alone“ Programm fungieren?
- Wie hoch sind die Kosten für die Programmierung, Implementierung und Wartung?
- Wie viel Zeit benötigt die Programmierung der Datenbankanwendung?
- Ist eine Vergleichbarkeit der Bedienoberflächen mit anderen Unternehmensanwendungen gegeben?

Im Folgenden werden auf Grundlage der abgebildeten Kriterien drei Programmierlösungen kurz vorgestellt. Diese haben sich durch eine Vorrecherche ergeben. Sie wurde per Internet- und Literatur-Recherche abgehandelt und soll an dieser Stelle nicht weiter vertieft werden. Herausgestellt haben sich die Programmierumgebungen SAP-ABAP, Visual C# in Kombination mit einem SQL-Server und Microsoft-Access.

4.1.1 SAP NetWeaver - ABAP

Bei der Programmiersprache ABAP handelt es sich um eine Entwicklung der Softwarefirma SAP. Die Sprache selbst ist nach [KSE07] eine Programmiersprache der 4. Generation. Das bedeutet, dass die benötigten Parameter für die Programmierung bereits in einzelnen Paketen enthalten sind. Diese müssen passend zusammengestellt werden und durch Eingabe von Programmcode wird eine individuelle Anpassung vorgenommen. Des Weiteren ist SQL vollständig in die Programmiersprache und die Programmierumgebung integriert. Somit handelt es sich um eine einfach lesbare Programmiersprache.

Der Aufbau einer SAP-ABAP-Programmdatenbank erfolgt in einer so genannten 3-Schicht-Architektur¹⁰. Diese besteht aus dem GUI (Graphical User Interface), dem Applikationsserver und der Datenbank. Das GUI stellt die Verbindung zum Anwender dar. Auf dem Applikationsserver laufen die Programme und es erfolgt die zentrale Steuerung. In der Datenbank werden lediglich die Daten abgelegt. Diese befindet sich auch zentral auf einem Server. Alle SAP-Systeme verfügen über eine Aufteilung in ein Produktivsystem sowie ein Test- und Qualitätssicherungssystem. In dem Test- und Qualitätssicherungssys-

¹⁰ Neben der 3-Schicht-Architektur gibt es noch die 1-, 2- und N-schichtige Architektur. Als Schicht wird der Ablageort für die Daten bezeichnet, aus denen die Anwendung besteht. Dieser Ablageort kann Zentral oder Dezentral im Unternehmen sein.

tem können Änderungen vorgenommen und überprüft werden, ohne dass es Auswirkungen auf das Produktivsystem hat. Somit ist sichergestellt, dass nur eine lauffähige und getestete Anwendung in das Produktivsystem überstellt wird.

Wie bei anderen Datenbankanwendungen wird auch hier eine Informationsverteilung über Tabellen vorgenommen. In ihnen werden die Daten später gespeichert. Die Speicherung selbst erfolgt in diesem Fall in einer so genannten „SAP maxDB“. Dabei handelt es sich um eine Datenbank von SAP, für die keine zusätzlichen Lizenzkosten anfallen würden. Verknüpft werden diese Tabellen über Fremdschlüssel¹¹.

Die Programmierung der Benutzeroberfläche erfolgt über vorgegebene Befehle. In der Abbildung 5 ist ein Selektionsbild aus dem SAP Beispielprogramm zu sehen. Dieses Programm dient der Einführung in die Programmiermöglichkeiten. Es orientiert sich an einer Flugdatenverwaltung. Anhand solcher grafischen Oberflächen erfolgt die Eingabe und Auswertung von Daten innerhalb der Datenbank. In der Abbildung ist ebenfalls zu erkennen, dass sich eine Standardkopfzeile immer innerhalb der Oberflächen befindet. Anhand dieser ist eine gewohnte Navigation der Anwendung möglich.

Abbildung 5: Selektionsbild aus dem SAP Beispielprogramm

Die Ergebnisse einer Abfrage beziehungsweise Selektion werden in einer Tabelle wiedergegeben. Dargestellt werden diese Inhalte in sogenannten Forms. Dieses ist in Abbildung 6 zu sehen. Zusätzlich stehen an dieser Stelle bereits standardmäßig Exportfunktionen zur Verfügung, beispielsweise zu Word, Excel und Outlook. Auch Druckfunktionen zu einem Ausgabegeräte oder eine Konvertierung in eine PDF-Datei¹² sind möglich. Diese Funktionen sind am oberen Rand der Tabelle in der Abbildung 6 zu sehen.

Als weitere Funktionen sind an dieser Stelle auch noch Möglichkeiten der Filterung und Sortierung gegeben. All diese gehören laut [KK06] bereits zu den Standardfunktionen von SAP NetWeaver und sind in ABAP enthalten. Die Programmierung und auch die spätere Nutzung der Datenbankanwendung erfolgt über die SAP-Oberfläche oder über einen Webbrowser. Dieses ermöglicht auch eine mobile Nutzung der Anwendung.

¹¹ Über Fremdschlüssel werden Beziehungen zwischen einzelnen Tabellen hergestellt. Näher erläutert wird diese in Kapitel 5.4.5.

¹² PDF ist ein Dateiformat der Firma Adobe. Es dient der Darstellung von Dokumenten.

Die Programmierung und Wartung würde bei der Salzgitter Flachstahl GmbH zentral durch eine IT-Abteilung erfolgen. Somit ist eine kostenneutrale Umsetzung innerhalb der Abteilung des themengebenden Unternehmens nicht möglich. Eine zentrale Programmierung würde etwa fünf Werktage für einen erfahrenen Programmierer in Anspruch nehmen. Erst dann wären alle Vorgaben auch umgesetzt. Spätere Änderungen an der Anwendung sind, bei Wahl dieser Programmierungsumgebung, nicht durch die einsetzende Abteilung umsetzbar. Zusätzlich zu den Programmierkosten fallen auch laufende Lizenzkosten für die Nutzung von SAP an. Da später nicht nur Anwender mit bereits vorhandenem SAP-Zugang die Datenbankanwendung verwenden sollen, würden hier zusätzliche Kosten anfallen. So würden für bereits vorhandene SAP Nutzer keine weiteren Lizenzkosten auflaufen, nur für hinzukommende Nutzer.

| ID | Nr | Lnd | Abflugstadt | Star | Lnd | Ankunftstadt | FLH | Flugd. | Abflugzeit | Ankunft | Entfernung | Entr | C | Ta |
|----|------|-----|---------------|------|-----|---------------|-----|--------|------------|----------|------------|------|---|----|
| AA | 17 | US | NEW YORK | JFK | US | SAN FRANCISCO | SFO | 6:01 | 11:00:00 | 14:01:00 | 2.572 | MI | | 0 |
| AA | 64 | US | SAN FRANCISCO | SFO | US | NEW YORK | JFK | 5:21 | 09:00:00 | 17:21:00 | 2.572 | MI | | 0 |
| AZ | 555 | IT | ROME | FCO | DE | FRANKFURT | FRA | 2:05 | 19:00:00 | 21:05:00 | 845 | MI | | 0 |
| AZ | 788 | IT | ROME | FCO | JP | TOKYO | TYO | 12:55 | 12:00:00 | 08:55:00 | 6.130 | MI | | 1 |
| AZ | 789 | JP | TOKYO | TYO | IT | ROME | FCO | 15:40 | 11:45:00 | 19:25:00 | 6.130 | MI | | 0 |
| AZ | 790 | IT | ROME | FCO | JP | OSAKA | KIX | 13:35 | 10:35:00 | 08:10:00 | 6.030 | MI | X | 1 |
| DL | 106 | US | NEW YORK | JFK | DE | FRANKFURT | FRA | 7:55 | 19:35:00 | 09:30:00 | 3.851 | MI | | 1 |
| DL | 1699 | US | NEW YORK | JFK | US | SAN FRANCISCO | SFO | 6:22 | 17:15:00 | 20:37:00 | 2.572 | MI | | 0 |
| DL | 1984 | US | SAN FRANCISCO | SFO | US | NEW YORK | JFK | 5:25 | 10:00:00 | 18:25:00 | 2.572 | MI | | 0 |
| JL | 407 | JP | TOKYO | NRT | DE | FRANKFURT | FRA | 12:05 | 13:30:00 | 17:35:00 | 9.100 | KM | | 0 |
| JL | 408 | DE | FRANKFURT | FRA | JP | TOKYO | NRT | 11:15 | 20:25:00 | 15:40:00 | 9.100 | KM | X | 1 |
| LH | 400 | DE | FRANKFURT | FRA | US | NEW YORK | JFK | 7:24 | 10:10:00 | 11:34:00 | 6.162 | KM | | 0 |
| LH | 401 | US | NEW YORK | JFK | DE | FRANKFURT | FRA | 7:15 | 18:30:00 | 07:45:00 | 6.162 | KM | | 1 |
| LH | 402 | DE | FRANKFURT | FRA | US | NEW YORK | JFK | 7:35 | 13:30:00 | 15:05:00 | 6.162 | KM | X | 0 |
| LH | 2402 | DE | FRANKFURT | FRA | DE | BERLIN | SXF | 1:05 | 10:30:00 | 11:35:00 | 555 | KM | | 0 |
| LH | 2407 | DE | BERLIN | TXL | DE | FRANKFURT | FRA | 1:05 | 07:10:00 | 08:15:00 | 555 | KM | | 0 |
| QF | 5 | SG | SINGAPORE | SIN | DE | FRANKFURT | FRA | 13:45 | 22:50:00 | 05:35:00 | 10.000 | KM | | 1 |
| QF | 6 | DE | FRANKFURT | FRA | SG | SINGAPORE | SIN | 11:10 | 20:55:00 | 15:05:00 | 10.000 | KM | | 1 |
| SQ | 2 | SG | SINGAPORE | SIN | US | SAN FRANCISCO | SFO | 18:25 | 17:00:00 | 19:25:00 | 8.452 | MI | | 0 |
| SQ | 15 | US | SAN FRANCISCO | SFO | SG | SINGAPORE | SIN | 18:45 | 16:00:00 | 02:45:00 | 8.452 | MI | | 2 |
| SQ | 158 | SG | SINGAPORE | SIN | ID | JAKARTA | JKT | 1:35 | 15:25:00 | 16:00:00 | 560 | MI | | 0 |
| SQ | 988 | SG | SINGAPORE | SIN | JP | TOKYO | TYO | 6:40 | 16:35:00 | 00:15:00 | 3.125 | MI | | 1 |
| UA | 941 | DE | FRANKFURT | FRA | US | SAN FRANCISCO | SFO | 11:36 | 14:30:00 | 17:06:00 | 5.685 | MI | | 0 |
| UA | 3504 | US | SAN FRANCISCO | SFO | DE | FRANKFURT | FRA | 10:30 | 15:00:00 | 10:30:00 | 5.685 | MI | | 1 |
| UA | 3516 | US | NEW YORK | JFK | DE | FRANKFURT | FRA | 7:25 | 16:20:00 | 05:45:00 | 6.162 | KM | | 1 |
| UA | 3517 | DE | FRANKFURT | FRA | US | NEW YORK | JFK | 8:15 | 10:40:00 | 12:55:00 | 6.162 | KM | | 0 |

Abbildung 6: Selektionsergebnisse aus dem SAP Beispielprogramm

Mit der Programmiersprache ABAP ist eine objektorientierte¹³ oder prozedurale¹⁴ Programmierung umsetzbar. Bereits bei der Programmierung ist es möglich, Nutzerrechte zu vergeben. So lässt sich eine individuelle Anpassung der Rechte an der Datenbankanwendung vornehmen. Diese Nutzerrechte werden in so genannten „Rollen“ definiert. Eine

¹³ Bei der objektorientierten Programmierung erfolgt der Softwareentwurf in einzelnen Objekten - den Tabellen. Näheres siehe Kapitel 5.2.2.

¹⁴ Bei der prozeduralen Programmierung wird die Gesamtaufgabe nach [PH09], die eine Software lösen soll, in kleine Teilaufgaben aufgelöst. Die dabei entstehenden Programm-Module werden Prozeduren genannt. Jede Teilaufgabe für sich lässt sich einfacher beschreiben, programmieren und testen.

Erweiterung um Sonderfunktionen ist in fast jeder Form möglich. Die Programmierung wird zu jeder Zeit von einer umfangreichen Fehlerdiagnose gestützt. So ist bei SAP die Abfragesprache vollständig in die Programmiersprache integriert. Dadurch fallen Fehler nicht erst bei der Ausführung der Datenbankanwendung auf, sondern bereits bei der Programmierung.

Im laufenden Programm können Archivierungsfunktionen zur Tabellenleerung implementiert werden. Somit wird ein zu großer Datenbestand unterbunden. Automatismen zur Auftragsvergabe aus der Anwendung heraus können ebenfalls integriert werden. Mit diesen ist es möglich, an andere Programme Arbeitsaufträge zu erteilen oder Mitarbeiter zu benachrichtigen. Eine Benutzersteuerung ist ebenfalls implementierbar. So kann eine Mehrzahl von Personen gleichzeitig auf die Datenbank zugreifen und mit dieser arbeiten. Es wird dabei immer nur der Datensatz blockiert, der bereits durch einen anderen Benutzer bearbeitet wird. Es könnten sich aber jederzeit alle Datensätze einer Datenbank angesehen werden. Für eine tiefer reichende Erforschung oder für einen ersten Einblick in SAP NetWeaver mit ABAP kann eine Trial-Version mit einer Testfunktion über 30 Tage bei SAP heruntergeladen werden.

4.1.2 Visual C# mit SQL-Server

Die Programmiersprache C#¹⁵ wurde von Microsoft entwickelt. Es handelt sich laut [AA13] um eine objektorientierte Sprache. Die aktuelle Version 5.0 wurde im Jahre 2012 herausgegeben. Die Programmierung erfolgt über einen Programmiereditor. Im Folgenden wird sich auf die Programmierung mit dem Programmiereditor „Microsoft Visual Studio 2010“ bezogen.

Bei einer Programmierung, die mit C# und einem Microsoft-SQL-Server realisiert wird, handelt es sich um eine 2-schichtige Datenbankanwendung. Gemäß [G11] bedeutet dies, dass die Präsentationsschicht (Benutzerschnittstelle) und die Geschäftsschicht (Datenbankanwendungssteuerung) auf einem Client laufen und die Datenbankschicht (Ablage der Datenbank) auf einem Datenbankserver. Bei der Programmierung würde die Benutzerschnittstelle und die Programmsteuerung in C# realisiert werden. Diese wird zu einem lauffähigen Programm konvertiert und kann auf einem Benutzer-PC (Client) ausgeführt werden. Das Programm wird über SQL¹⁶ an einen Datenbankserver angebunden, auf welchen sich die Datenbank befindet. So besteht diese Programmierlösung aus zwei Komponenten. Die Schnittstelle zwischen Benutzeroberfläche und Datenbank erfolgt via „ado.net“, mit ihrer Hilfe erfolgt das objektrelationale Mapping¹⁷. Dabei wird die Daten-

¹⁵ C# kommt aus dem Englischen und wird „C-Sharp“ gesprochen.

¹⁶ SQL (Structured Query Language) ist eine Datenbanksprache zur Definition von Datenstrukturen sowie zum Bearbeiten (Einfügen, Verändern, Löschen) und Abfragen von darauf bestehenden Datenbeständen.

¹⁷ Objektrelationale Abbildung (mapping) ist eine Technik der Softwareentwicklung, mit der ein in einer objektorientierten Programmiersprache geschriebenes Anwendungsprogramm seine Objekte in einer relationalen Datenbank ablegen kann.

bank-Tabellenstruktur auf die objektorientierte Struktur der C# Anwendung abgebildet. Es handelt sich also um die Datenbank-Zugriffstechnologie.

Zunächst erfolgt die Erstellung der Tabellen für den SQL-Datenbankserver. Wie bei der SAP-Lösung werden auch hier die Daten zu Tabellen strukturiert. Somit wird eine gute Übersichtlichkeit erzeugt. Die Steuerung der Abhängigkeiten zwischen den Tabellen erfolgt aber über die Programmoberfläche, welche mit C# generiert wird. Eine direkte Verbindung über Schlüssel ist bei dieser Lösung nicht möglich. Wie in Abbildung 7 zu sehen ist, wird die Benutzeroberfläche via „Drag & Drop“¹⁸ erstellt. Ein benötigtes Objekt wird von der Toolbox (links im Bild) einfach auf eine Oberfläche gezogen. Dabei wird das Objekt mit einem Namen versehen (rechts unten im Eigenschaften-Fenster zu sehen), dieser wird in einer späteren Programmierung wieder herangezogen. Auf diese Art und Weise kann auch die Anbindung an die Datenbank erfolgen. Dafür wird mit dem „Server-Explorer“ (links im Bild) eine geeignete Oberfläche erstellt. In der Abbildung 7 ist ein Teil der Oberfläche einer DVD-Datenbank zu sehen. Diese wurde durch den Autor im Laufe seines Studiums erstellt. Im Projektmappen-Explorer ist eine Übersicht der enthaltenen Projektoberflächen ersichtlich.

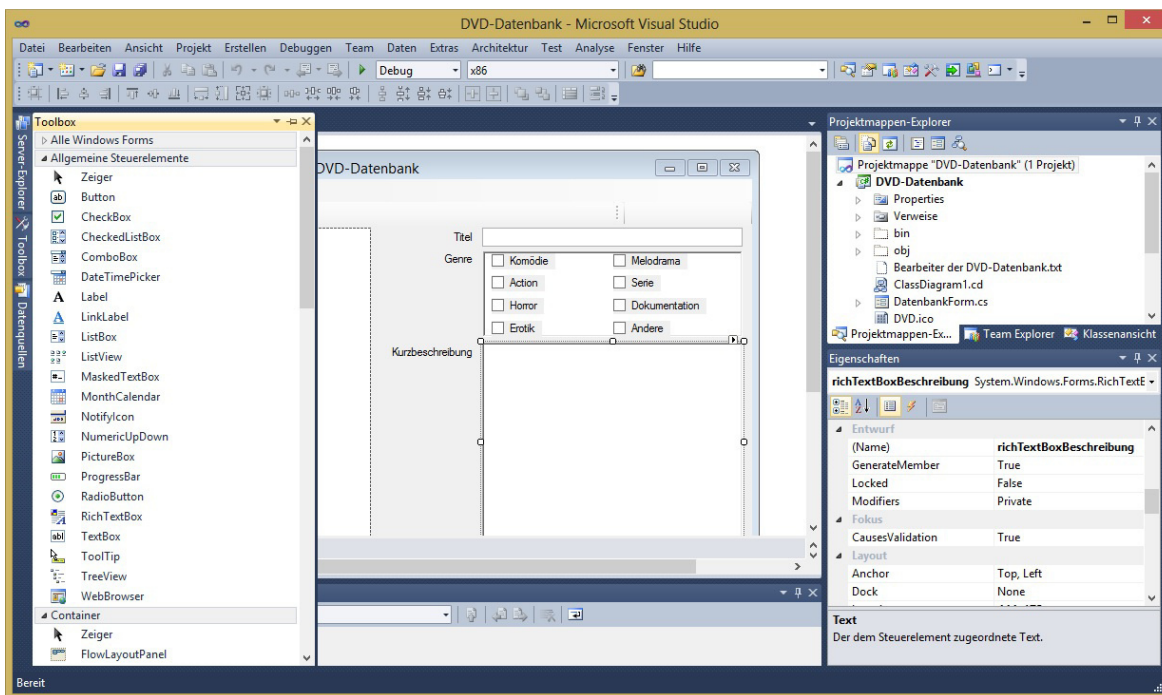


Abbildung 7: Erstellung der Benutzeroberfläche unter C#

Konvertierungen zu anderen Programmen wie beispielsweise Word, Excel und Adobe PDF sind nach [K12] nur sehr aufwendig zu implementieren. Es gibt hier keine vorgefertigten Programmbausteine. An dieser Stelle müssen die Möglichkeiten in einzelnen Schritten programmiert werden. Für einen Versand von E-Mails gibt es wiederum Programmierbefehle unter C#. Mit ihrer Hilfe ist eine solche Funktion erstellbar.

¹⁸ Zu Deutsch „Ziehen und Ablegen“, eine gängige Methode zum Erstellen grafischer Oberflächen.

In der nachfolgenden Abbildung 8 wird die Anbindung an eine Access-Datenbank dargestellt. Dabei sind auch die Befehle für den Datenaustausch zu sehen. Es handelt sich um SQL-Befehle. Dieses sind „update“, „insert“ und „delete“. „Bei SQL wird dem Datenbankmanagement-System mitgeteilt, was das Ergebnis sein soll und nicht wie die Aufgabe gelöst wird“ [G11]. Im oberen Teil der Abbildung ist die Anbindung an die Datenbank selbst zu sehen. Dieses geschieht über den eingeblendeten String und den Befehl „con.Open()“.

```
//Datenbankverbindung aufbauen
string s = @"Provider=Microsoft.Jet.OLEDB.4.0; Data Source = dvdDatenbank.mdb";
OleDbConnection con = new OleDbConnection(s);
con.Open();
if (con.State == ConnectionState.Open)
{
    toolStripStatusLabel1.ActiveLinkColor = Color.Green;
    toolStripStatusLabel1.Text = "Datenbank erfolgreich eingebunden!";
    toolStripStatusLabel1.ForeColor = Color.Green;
}
else
{
    toolStripStatusLabel1.Text = "Datenbank nicht eingebunden!";
}
//Data Adapter anlegen
da = new OleDbDataAdapter("select * from DatenInhalte", con);
//Data Adapter konfigurieren
OleDbCommand cmdupdate = new OleDbCommand("UPDATE DatenInhalte set Titel=?, Genr
OleDbCommand cmdinsert = new OleDbCommand("INSERT INTO DatenInhalte (Titel, Genr
OleDbCommand cmddelete = new OleDbCommand("DELETE FROM DatenInhalte where ID = ?

da.UpdateCommand = cmdupdate;
da.InsertCommand = cmdinsert;
da.DeleteCommand = cmddelete;
```

Abbildung 8: Quellcode für die Anbindung an eine Access-Datenbank

Für die Darstellung von Abfrageergebnissen müsste eine eigene Maske entwickelt werden, mit deren Hilfe auch unter C# eine Auswertung möglich ist. Filterfunktionen oder Ähnliches sind hier standardmäßig nicht mit dabei. All diese Features können über zusätzliche Programmierungen implementiert werden, was natürlich mit einem Extra an Aufwand verbunden ist. Für die Abfrage selbst gibt es entsprechende Befehle, die in SQL enthalten sind. So kann beispielsweise mit dem Befehl „select“ ein Datenbestand aus einer Tabelle innerhalb der Datenbank abgefragt werden.

Bei C# sind die Befehle von SQL nicht vollständig in die Programmiersprache integriert. Das bedeutet, dass Programmierfehler erst bei einem Test des Programms auffallen und nicht schon bei der Programmierung in Visual Studio bemängelt werden. Dieses kann zu zusätzlichem Aufwand bei dem Test und der Implementierung der Anwendung führen.

Eine Steuerung der Zugriffsrechte für mehrere Benutzer ist ebenfalls möglich. Entsprechende Rechte können über Programmierbefehle integriert werden. Bei einer Erneuerung der Benutzerrechte müssten aber alle mit C# erstellten Client-Programme erneuert werden, da sonst die Nutzerrechte nicht übertragen sind. Allgemein ist eine Programmerneuerung möglich, aber nur, wenn die Programme händisch durch den Nutzer oder Sys-

temadministrator auf jedem Client-PC erneuert werden. Durch die 2-Schicht-Architektur ist eine zentrale Erneuerung der Benutzeroberfläche nicht gegeben. Die Geschäftsschicht der Anwendung liegt in diesem Fall nicht auf dem Server, sondern auf dem Client. Dieser Nachteil erhöht den Wartungsaufwand immens.

Lizenzkosten fallen bei dieser Lösungsmöglichkeit einmalig für die Programmierumgebung Visual Studio an. Da die Lösung ein eigenständiges Programm hervorbringt und auch der SQL-Server keine Lizenzkosten verursacht (eine lizenzfreien Variante ist auch in Unternehmen verwendbar), fallen keine laufenden Kosten in diesem Bereich an. Die Datenbankanwendung selbst kann vom Autor programmiert werden. Somit belaufen sich die Kosten für die Programmierung lediglich auf abteilungsinterne Kosten. Eine Beauftragung der zentralen Wartung ist ebenfalls nicht nötig. Auch diese Aufgaben könnten abteilungsintern verteilt werden. Somit lassen sich auch die externen Kosten für den laufenden Betrieb fast auf Null reduzieren. Lediglich die Datenbank selbst sollte zentral auf einem Server untergebracht werden. Die Dauer für die Erstellung der Datenbankanwendung wird auf circa acht Arbeitstage für einen erfahrenen Programmierer geschätzt.

4.1.3 Microsoft Access

Das Programm Access ist eine Datenbanklösung der Firma Microsoft. In diesem sind alle Bausteine für eine umfassende Datenbank integriert. Daher handelt es sich dabei auch um eine so genannte 1-schichtige Datenbankanwendung. Das bedeutet, dass sich sowohl die Präsentationsschicht, die Geschäftsschicht als auch die Datenbankschicht in einer Oberfläche befinden und an einem gemeinsamen Speicherort abgelegt sind. In den meisten Fällen wird dieser lokal auf einem Client sein. Die Basis der Programmierung bildet ein relationales Datenbanksystem. Das bedeutet nach [H10], dass die Datenhaltung über Tabellen erfolgt, die in Beziehung (Relation) zueinander stehen.

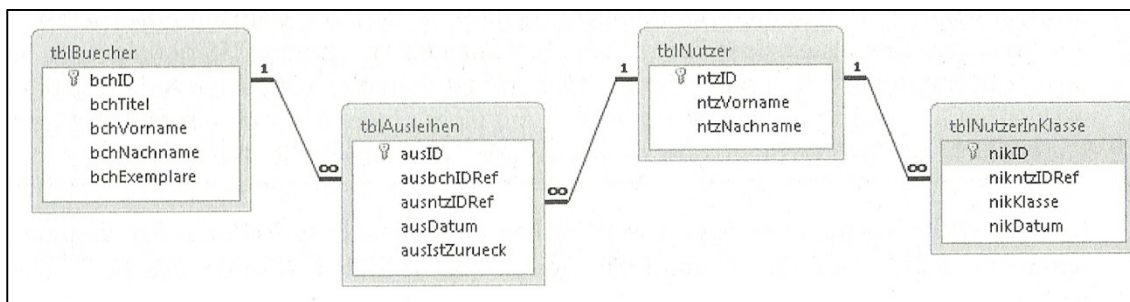


Abbildung 9: Tabellen einer Datenbank in Relation zueinander [H10 - S.122]

Bei Beginn der Programmierung einer Access-Datenbank werden zunächst Tabellen gebildet. Dieses wird in der Abbildung 9 noch einmal verdeutlicht. Jede Tabelle wird mit einem Primärschlüssel¹⁹ versehen und es werden Beziehungen²⁰ zwischen den Tabellen

¹⁹ Ein Primärschlüssel dient der eindeutigen Identifizierung eines Tupels innerhalb einer Tabelle. Näher erläutert wird dieser im Kapitel 5.4.5.

²⁰ Bei Beziehungen zwischen Tabellen gibt es verschiedene Arten, sie in Relation zu setzen. Näher erläutert werden diese im Kapitel 5.4.

hergestellt. Dadurch werden die Abhängigkeiten innerhalb der Datenbank festgelegt. Die Primärschlüssel der Tabellen und die Art der Beziehungen sind ebenfalls in der Abbildung 9 wiederzufinden.

In den nächsten Schritten ist es möglich, eine Nutzermaske zu erstellen. Ein Beispiel dafür ist in der Abbildung 10 zu finden. Weitere Elemente der Programmierumgebung finden sich im linken Bildbereich der Abbildung. Dort ist eine Übersicht der vorhandenen Tabellen, Abfragen, Berichte und Formulare (Benutzermasken) angeordnet. Im oberen Bereich der Abbildung sind die Steuerelemente für die Erstellung der Datenbankanwendung angeordnet. Mit Hilfe dieser lassen sich alle Elemente per „Drag & Drop“ innerhalb der Oberflächen platzieren. Dadurch ist eine leichte Erstellung von Funktionen gegeben.

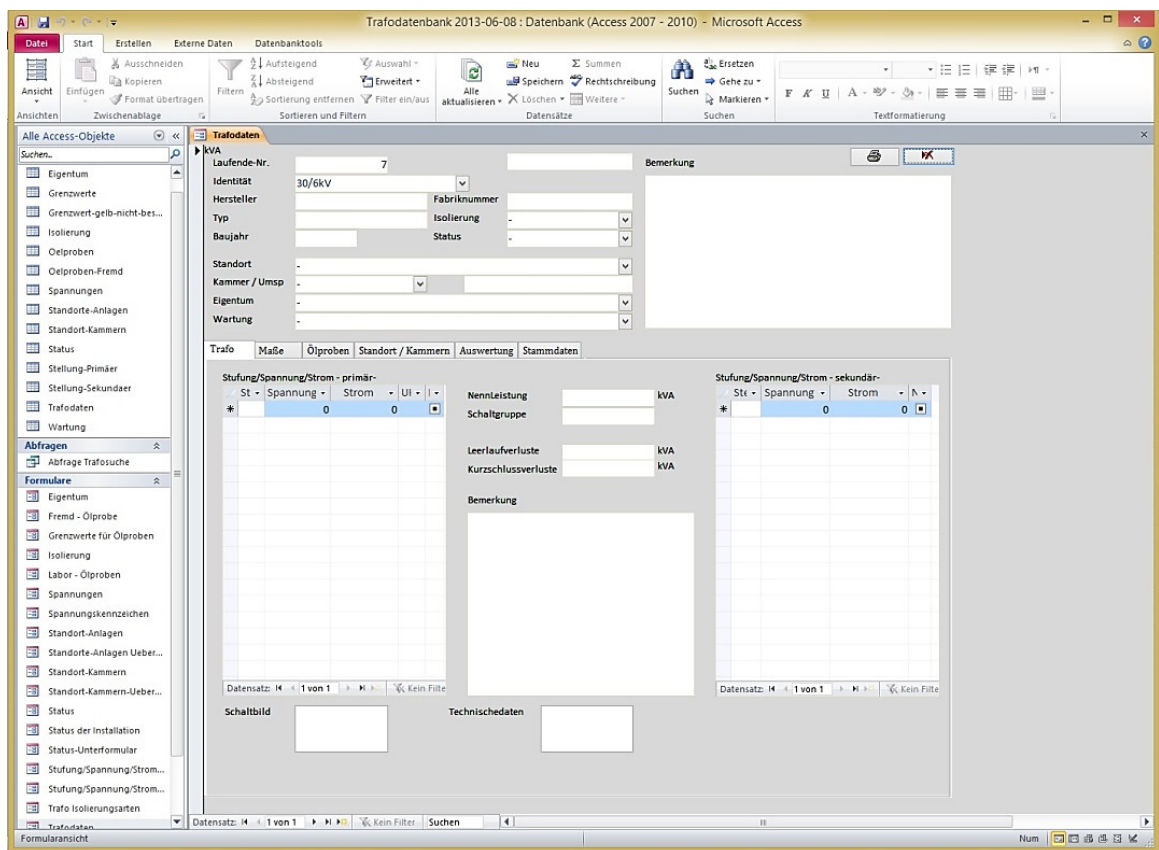


Abbildung 10: Übersicht der Access Programmoberfläche

Laut [H11] sind umfangreiche Funktionen via VBA programmierbar. So ist die Erstellung von anspruchsvollen Anwendungen möglich. Durch die Inkludierung aller Datenbankanwendungsbausteine in einem Programm ist auch die Abfragesprache vollständig in die Programmiersprache integriert. Jedoch ist für die grundsätzliche Erstellung einer Datenbankanwendung mit Access keine Programmiersprache Voraussetzung.

Abfragen lassen sich in so genannten Berichten wiedergeben. Mit Hilfe einer frei erstellbaren Berichtsoberfläche ist so eine individuelle Gestaltung möglich. Suchfunktionen und Filter sind ebenfalls im Grundpaket enthalten. Diese Bausteine lassen sich per Abfragefunktion umsetzen.

Konvertierungsfunktionen zu anderen Office-Anwendungen²¹ sind möglich. Wobei sich Konvertierungen zu Excel und Word am einfachsten umsetzen lassen, diese Funktionen sind vorinstalliert. Auch eine Druckausgabe von Datensätzen ist bereits als Paket mit im Funktionsumfang. Eine Konvertierung der Daten und Berichte direkt in das PDF-Format ist nicht standardmäßig enthalten und müsste aufwändig programmiert werden. Alternativ könnte über die Druckfunktion eine PDF-Datei erstellt werden.

Da es sich bei Access um eine 1-Schicht-Datenbankanwendung handelt, sind Aktualisierungen schwerer umzusetzen. Diese müssen vom Datenbankadministrator durchgeführt werden, um sie anschließend händisch auf alle Client-Plätze zu verteilen. Eine Automatisierung, wie bei einer 3-Schicht-Anwendung, ist hier nicht gegeben. Alternativ kann die Datenbank zentral auf einem Server abgelegt werden. Auf jedem Client könnte ein Link zur Datenbank installiert werden, mit dessen Hilfe die Anwendung aufgerufen wird.

Ein Minuspunkt ist die Vergabe von Nutzerrechten. Access hat an dieser Stelle lediglich einen Exklusiv-Modus vorgesehen. Damit ist es möglich die Datenbank mit einem Kennwort zu verschlüsseln, so dass nur bestimmte Teile zugänglich sind. Eine Unterscheidung von verschiedenen Rechten wird so aber nicht vorgenommen. Bei einer aufwendigen Programmierung in VBA sind Nutzergruppen erstellbar. Für sie können verschiedene Rechte (zum Beispiel nur lesen oder lesen/schreiben) vergeben werden. Ein Mehrbenutzerbetrieb ist nach [M11] möglich. Dabei muss aber die Datenbank in ein Backend und Frontend geteilt werden. Im Backend sind nur die Tabellen und im Frontend alle übrigen Objekte enthalten. Die Trennung erfolgt per Assistenten über die Access-Tools.

Abschließend werden die Lizenzkosten betrachtet. Diese belaufen sich auf die Lizenz für Microsoft Access. Beim themengebenden Unternehmen ist das Programm Access bereits im Standard-Office-Paket enthalten. Somit steht es an allen Client-Plätzen zur Verfügung und es entstehen keine weiteren Lizenzkosten. Die Wartung und Instandhaltung der Datenbankanwendung kann abteilungsintern vergeben werden. Damit belaufen sich die Kosten für die Unterhaltung der Datenbank gegen Null. Lediglich die Kosten für die Ablage der Anwendung auf einem Server sollten an dieser Stelle berücksichtigt werden. Für kleine Veränderungen oder Verbesserung werden voraussichtlich keine tiefgreifenden Programmierkenntnisse benötigt. Die Erstellung der Datenbankanwendung unter Access wird für einen erfahrenen Programmierer voraussichtlich fünf Arbeitstage in Anspruch nehmen.

4.2 Bewertung der Marktanalyse/Programmauswahl

Im Kapitel 4.1 werden drei Datenbank-Programmierungsumgebungen vorgestellt. Um die Marktanalyse abschließen zu können, werden im Folgenden die Ergebnisse zusammengefasst dargestellt. Dazu werden sie in die Gruppen „must have“, „can have“ und „nice to have“ unterteilt. Zu Deutsch bedeutet dies die Eingruppierung in Eigenschaften, welche

²¹ Office-Anwendungen sind Programme von Microsoft. Je nach Paketumfang können unter andrem die Programme Word, Excel, PowerPoint, Outlook, Access, Publisher, OneNote, InfoPath und Lync enthalten sein.

die Datenbank haben muss (must have), welche die Datenbank haben kann (can have) und die, bei denen es schön wäre, wenn die Datenbank diese aufweisen würde (nice to have). Die darzustellenden Eigenschaften stammen aus den Kapiteln 3.2 und 4.1. Aufgeteilt werden sie nach den oben genannten Kriterien in die folgenden drei Tabellen.

| <u>Datenbankeigenschaften - „must have“</u> | | | |
|-------------------------------------------------------------|---------------|--------------------|---------|
| Kriterium: | SAP – ABAP: | C# mit SQL-Server: | Access: |
| Eingabe von Daten über eine individuelle Benutzermaske | Erfüllt | Erfüllt | Erfüllt |
| Konvertierungsfunktionen zu Word und Excel | Erfüllt | Teilweise erfüllt | Erfüllt |
| Druckausgabe von Datenbeständen und Suchergebnissen | Erfüllt | Teilweise erfüllt | Erfüllt |
| Speicherung von externen Dokumenten in der Datenbank | Erfüllt | Erfüllt | Erfüllt |
| Programmierung und Betreuung der Datenbank abteilungsintern | Nicht erfüllt | Erfüllt | Erfüllt |
| Kostenneutralität für die Unterhaltung der Datenbank | Nicht erfüllt | Erfüllt | Erfüllt |
| Bedienung der Anwendung durch PC-Laien möglich | Nicht erfüllt | Erfüllt | Erfüllt |

Tabelle 2: Datenbankanwendungseigenschaften - must have

| <u>Datenbankeigenschaften - „can have“</u> | | | |
|----------------------------------------------------------|-------------|--------------------|-------------------|
| Kriterium: | SAP – ABAP: | C# mit SQL-Server: | Access: |
| Wartung servergesteuert mit automatischer Aktualisierung | Erfüllt | Nicht erfüllt | Nicht erfüllt |
| Ablage der Datenbank zentral auf einem Server | Erfüllt | Teilweise erfüllt | Teilweise erfüllt |
| Mehrbenutzerbetrieb möglich | Erfüllt | Teilweise erfüllt | Erfüllt |

Tabelle 3: Datenbankanwendungseigenschaften - can have

| <u>Datenbankeigenschaften - „nice to have“</u> | | | |
|---------------------------------------------------------------------------|-------------|--------------------|-----------------------------------------------|
| Kriterium: | SAP – ABAP: | C# mit SQL-Server: | Access: |
| Vergleichbarkeit der Bedienoberfläche mit anderen Unternehmensanwendungen | Erfüllt | Nicht erfüllt | Nicht erfüllt |
| Verknüpfbarkeit der Datenbank mit anderen Anwendungsprogrammen. | Erfüllt | Teilweise erfüllt | Teilweise erfüllt |
| Nutzung der Datenbank auf mobilen Endgeräten. | Erfüllt | Nicht erfüllt | Teilweise erfüllt (je nach Betriebssystem) |

Tabelle 4: Datenbankanwendungseigenschaften - nice to have

In den vorhergehenden drei Tabellen sind die Datenbankanwendungseigenschaften zu finden. Die Kriterien nach [A91] werden im Folgenden für die Entscheidungsfindung mit herangezogen. Diese sind Abgeschlossenheit, Mengenorientiertheit, Effizienz und Vollständigkeit. Neben den aufgeführten Kriterien wird auch der Programmieraufwand bei der Entscheidung berücksichtigt.

Die verschiedenen Kriterien sind für die Vorstellung beim themengebenden Unternehmen zusätzlich mit einer Gewichtung versehen wurden. Bei den Kriterien im Bereich „must have“ ist die Gewichtung am größten, in den anderen Bereichen entsprechend abfallend. Auf Basis dieser Gewichtung und der oben aufgeführten Kriterien wird die Entscheidung für die Programmierumgebung gefällt. Ausgewählt ist das Programm Microsoft Access. Dieses hat auch die meisten zutreffenden Kriterien in den vorhergehenden Tabellen.

Eine große Gewichtung bei der Entscheidungsfindung lag auf der abteilungsinternen Programmierung und Wartung der Datenbankanwendung. Dieser Vorteil besticht, da auch von anderen Mitarbeitern der programmeinsetzenden Abteilung Erweiterungen und Änderungen im Nachhinein vorgenommen werden können. Ein weiterer Vorteil liegt beim Wiedererkennungswert und der Nutzerakzeptanz. Es werden auch andere Access-Datenbank-Anwendungen in der Abteilung eingesetzt. Schließlich liegt bei der Inkludierung vorhandener Daten noch eine Arbeitserleichterung vor, da sich ein Teil dieser Daten bereits in einer Access-Anwendung befindet. So können die Daten später bei der Integration des Datenbestandes leichter verwendet werden. Alleine durch diese Vorteile sind die in den vorhergehenden Tabellen zu findenden Nachteile aufgewogen.

5 Systemkonzept

5.1 Grundlagen der Datenbanken

Datenbanken gewinnen in der heutigen Zeit immer mehr an Bedeutung. So wurden früher beispielsweise Bibliotheken über Karteikarten verwaltet. Bei einer stetig zunehmenden Datenmenge beziehungsweise Bestand an Büchern und Ausleihern ist das heute nicht mehr möglich. Für die Verwaltung solcher Datenmengen werden Datenbanken benötigt. Die Aufgabe einer Datenbank ist es, eine Datenorganisation bereit zu stellen. Mit Hilfe dieser Organisation muss es möglich sein,

- dem Benutzer Zugriff zu gespeicherten Daten zu geben,
- sicherzustellen, dass Daten nicht unberechtigt manipuliert werden,
- die interne Datenorganisation zu ändern, ohne dass der Benutzer seine Anwenderprogramme anpassen muss.

Die Datenorganisation muss nach [S09] zur Erfüllung dieser Aufgaben eine dauerhafte, selbständige, flexible und sichere Organisation der Daten ermöglichen. Dabei sind die grundsätzlichen Eigenschaften einer Datenbankanwendung:

- einfacher Aufbau → keine Redundanzen²²
- Trennung von Dateninhalten²³ und Applikation²⁴
- Flexibilität → neue Applikationen ermöglichen
- Datenintegrität → keine widersprüchlichen Daten

Ermöglicht und gesteuert werden diese Funktionen über ein Datenbank-Management-System (DBMS). Das passende System zur gewählten Anwendung Access wird im Kapitel 5.2.4 näher erläutert.

5.2 Datenbankmodelle

In der Datenbankwelt wird eine Vielzahl an Verwaltungsmodellen eingesetzt. Im Detail werden diese in den Kapiteln 5.2.1 bis 5.2.4 näher beschrieben. Zunächst werden die Funktionen einer Datenverwaltung vorgestellt.

²² Keine Redundanzen impliziert, dass keine Daten doppelt vorhanden sein dürfen. Näher erläutert wird dieses im Kapitel 5.5.

²³ Dateninhalte werden bei einem relationalen Datenbankmodell in Tabellen gespeichert.

²⁴ Auf Applikationen wird bei einem relationalen Datenbankmodell über die Benutzeroberfläche zugegriffen.

Diese müssen sein:

- Eingabe von Daten
- Nachführen von Daten
- Löschen von Daten
- Abfrage bestehender Daten
- Schutz vor Datenverlust

Diese Aufgaben obliegen dem Datenbank-Managementsystem, wie auch schon im Kapitel 5.1 erwähnt. Nach [S09] lässt sich ein Datenbanksystem anhand eines Schalenmodells am besten erklären. In Abbildung 11 ist das Schalenmodell eines relationalen Datenbank-Management-Systems (RDBMS) abgebildet.

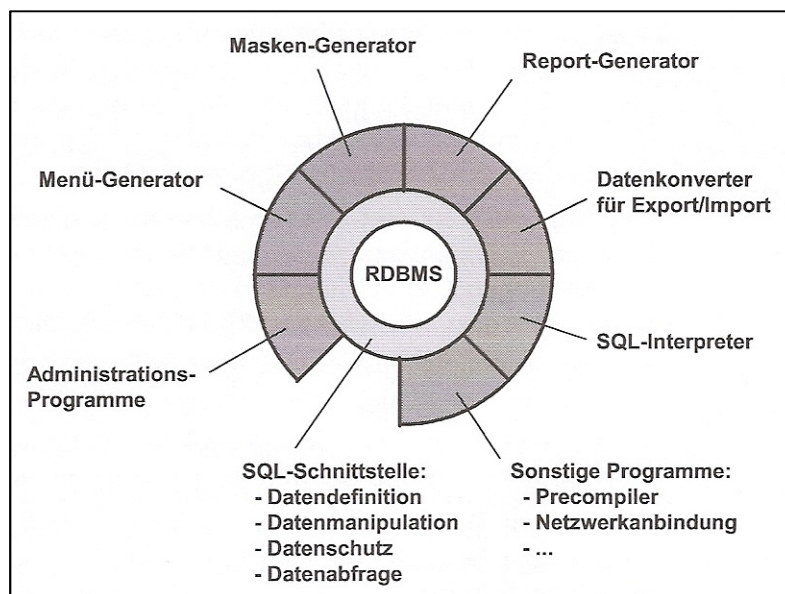


Abbildung 11: Schalenmodell eines RDBMS nach [S09 - S. 8]

Wie zu sehen ist, besteht der Kern einer Datenbank aus dem Management-System. In diesem speziellen Fall aus einem RDBMS. In dessen Kern sind die Systemfunktionen (suchen, lesen, schreiben, löschen von Daten) enthalten. SQL definiert die Schnittstelle des RDBMS. Die SQL-Strukturanteile sind:

- DDL → Data Definition Language – Datendefinition: Aufbau von Tabellen und Feldern
- DML → Data Manipulation Language – Datenmanipulation: Verändern (Nachführen) von Daten; Datensätze: eingeben, bearbeiten, löschen
- DSL → Data Security Language – Datenschutz: Zugriffsrechte und Datenschutz
- DRL → Data Retrieval Language – Datenabfrage: Selektion der Daten

Mit Hilfe der eben aufgeführten Funktionen erfolgt die Kommunikation zur äußeren Schicht des Schalenmodells. In ihr sind die Generatoren und Konverter untergebracht. Mit diesen erfolgt wiederum die Kommunikation zum Benutzer der Datenbank. Über den Menügenerator kann eine Auswahl von Eingabemasken oder Reports erfolgen. Der Masken-

generator dient der Erstellung von Eingabemasken und dazugehörigen Funktionen. Durch die zentrale Nutzung von Eingabemasken ist die Datenintegrität²⁵ sichergestellt. Mit Hilfe des Reportgenerators können Abfragen erstellt werden. Für einfache Abfragen kann ein SQL-Befehl genutzt werden, bei umfangreicheren oder sich wiederholenden Abfragen kann ein Report generiert werden.

Die Administrations-Programme dienen der Erstellung, Wartung und Reparatur einer Datenbank. Der Export und Import von Daten erfolgt über den Datenkonverter. So können Daten aus einer Datenbank auch anderen Anwendungen zur Verfügung gestellt werden. Zusammenfassend lässt sich nach [SW07] sagen, dass ein RDBMS die Gesamtheit aller Programme zur Erzeugung, Verwaltung und Manipulation einer Datenbank darstellt.

5.2.1 Das hierarchische Datenbankmodell

In einer hierarchischen Datenbank werden die Daten in einer einzelnen, sequenziellen Datei gespeichert. Dieses ist in Abbildung 12 dargestellt.

| | |
|-------|-------------|
| 1 | 134 |
| | Müller |
| 1.1 | 101 |
| | Schmiedeweg |
| 1.1.1 | 01.01.2003 |
| 1.2 | 102 |
| | Dunklestr. |
| 1.2.1 | 10.03.2004 |
| 1.2.2 | 01.02.2006 |
| 2 | 257 |
| | Schulz |
| 2.1 | 101 |
| | Schmiedeweg |
| 2.1.1 | 03.07.2007 |

Abbildung 12: Aufbau einer hierarchischen Datenbank

Die erste Spalte der Abbildung 12 gibt die Hierarchiestufe an. In der zweiten Spalte stehen die zugehörigen Daten. In dem aufgeführten Beispiel hat der Name die höchste Hierarchie. Danach folgen die Straße und das Datum. Wie sich an diesem Beispiel erkennen lässt, ist das Nachfügen von Daten sehr aufwendig. Es würde ein großes Verschieben innerhalb der Tabelle zur Folge haben. Daher werden neue Daten an das Ende der Tabelle gestellt und es wird mit Zeigern auf die jeweilige Stelle verwiesen. Trotz der umständlichen Handhabung ist dieses Modell nach [SSH10] neben dem netzwerkförmigen Modell das erfolgreichste Datenbankmodell der ersten Generation. Heute fristet diese Art von Datenbank ein Nischendasein.

²⁵ Die Datenintegrität umfasst die Maßnahmen, die sicherstellen, dass die Daten während der Übertragung oder Verarbeitung nicht beschädigt oder verändert werden.

5.2.2 Das objektorientierte Datenbankmodell

Objektorientierte Datenbanken bestehen aus Objekten statt Tabellen, wie bei den relationalen Datenbanken. Ein Objekt selbst kann nach [G11] aus Tabellen (Entitäten) und Beziehungen zwischen den Tabellen bestehen. Es werden also die einzelnen Objekte zu Themenkreisen sortiert und logisch miteinander verknüpft. Als weiterer Punkt ist auch der jeweilige Code²⁶ mit in den Objekten enthalten. Dieser Code dient der Verarbeitung der Objekte. Dadurch ist garantiert, dass das Objekt seine Eigenständigkeit und Wiederverwendbarkeit behält.

Die objektorientierten Datenbanken sind nach [S09] zur Zeit kaum verbreitet. Sie gewinnen aber einen immer größer werdenden Zuspruch. Das liegt an den komplexer werdenden Anforderungen an eine Datenbank. Diese können von relationalen Datenbanken nicht mehr bewältigt werden. Gerade in komplexen Anforderungen liegen die Stärken von objektorientierten Datenbanken. Ein weiteres Argument für die Verwendung von objektorientierten Datenbanken ist die in letzter Zeit gestiegene Verbreitung der objektorientierten Programmierung. Dadurch entsteht kein Bruch mehr in der Programmierung (mengenorientiert (relational) zu objektorientiert (Objekte)). Auf eine tiefergreifende Erläuterung zu diesem Datenbankmodell wird an dieser Stelle verzichtet, da es für die Programmierung der Datenbankanwendung nicht verwendet wird. Weiterführende Literatur in diesem Themengebiet ist beispielsweise [G11].

5.2.3 Das netzwerkförmige Datenbankmodell

Die netzwerkförmigen Datenbanken stellen eine Erweiterung der hierarchischen Datenbanken dar. Mit ihnen ist es möglich, bei den netzwerkförmigen Datenbanken mehr Beziehungstypen²⁷ zuzulassen. Eingesetzt werden diese bei Erweiterungen von bestehenden Datenverwaltungssystemen. Greift zum Beispiel eine Personalverwaltung auf einen Datenbestand zu, so ist es möglich, dass dieser Datenbestand auch für ein weiteres System genutzt wird. So lässt sich beispielsweise unter Verwendung eines Teils der Daten aus der Personalverwaltung eine Endgeldabrechnung hinzufügen. Statt aber eine neue Datenbank zu eröffnen, kann auch eine netzwerkförmige Datenbank installiert werden. Diese hat einen eigenen Datenbestand und ergänzt ihr System mit Daten aus der Personalverwaltung. So werden unnötige Datenanhäufungen vermieden, Speicherplatz gespart und die Systeme benötigen keine gesonderte Aktualisierung.

Die Schwächen dieses Systems sind ebenfalls ähnlich denen der hierarchischen Datenbanken. Auch hier muss zu den Daten navigiert werden, auf die zugegriffen werden soll. Dem Programmierer einer solchen Datenbank muss also die komplexe Struktur bekannt sein. Daraus ergibt sich eine strukturelle Abhängigkeit. Wenn die Struktur der Datenbank geändert wird, muss auch die Struktur der Anwenderprogramme angepasst werden. Tie-

²⁶ Ein Code ist eine Vorschrift zur Umwandlung von Nachrichten oder Befehlen, für die Übersetzung in ein Zielsystem. Im Allgemeinen ist ein Code eine Vereinbarung über einen Satz (eine Menge) von Zeichen.

²⁷ Die verschiedenen Beziehungstypen werden in Kapitel 5.4 näher erläutert.

fergreifende Erklärungen zu den netzwerkförmigen Datenbanken sind ebenfalls in [G11] zu finden.

5.2.4 Das relationale Datenbankmodell

Der Begriff des Entity-Relationship-Modells (kurz ER-Modell) geht zurück auf einen grundlegenden Artikel von P. P. Chen im Jahre 1976 [C76]. Seit dieser Zeit hat sich das Modell fest im Bereich der Datenbankmodelle verankert und wird nach [SSH10] – in abgewandelter Form – heutzutage faktisch als Standardmodell für frühe Entwurfsphasen der Datenbankentwicklung verwendet. Entwickelt wurde das ursprüngliche relationale Datenmodell nach [G11] von DR. E. F. Codd im Jahre 1970. Somit gilt er als der Erfinder des ER-Modells.

Der größte Vorteil der relationalen Datenbanken liegt darin, dass durch das RDBMS eine komplette Integration der physischen Daten erreicht wird. So kann sich der Anwender einzig mit der logischen Struktur der Datenbank beschäftigen, ohne sich Gedanken um die physikalische Speicherung machen zu müssen. Es wird also nur eine Art Nutzeroberfläche betrachtet und Abläufe bei einer Datenspeicherung (wo und wie sie gespeichert werden) müssen nicht gesondert betrachtet werden. Das ist ein großer Vorteil gegenüber den netzwerkförmigen und hierarchischen Datenbanken.

Wie bereits angeführt, erfolgt die Speicherung der Daten bei relationalen Datenbanken in Tabellen (Entitäten). Ein Beispiel dafür ist in Abbildung 13 zu sehen.

| Anwohner | | Straße | |
|----------|--------|--------|-------------|
| AW-Nr | Name | S-Nr | Straßenname |
| 134 | Müller | 101 | Schmiedeweg |
| 257 | Schulz | 102 | Dunklestr. |
| 313 | Meyer | 110 | Kaunenstr. |

| Einzugstermine | | |
|----------------|------|--------------|
| AW-Nr | S-Nr | Einzugsdatum |
| 134 | 101 | 01.01.2003 |
| 257 | 101 | 03.07.2007 |
| 257 | 102 | 21.06.2008 |
| 313 | 110 | 18.09.2010 |
| 313 | 101 | 21.03.2012 |

Abbildung 13: Tabellenorganisation bei relationalen Datenbanken

Wie in der Abbildung zu erkennen ist, erfolgt die Aufteilung der Tabellen nach Themenkreisen. In dem abgebildeten Beispiel gibt es je eine Tabelle für die Anwohner, die Straße und den Einzugstermin. Das ermöglicht eine wesentlich größere Flexibilität bei der Anwendung und Strukturergänzung. Bei letzterem würde einfach eine neue Tabelle hinzugefügt beziehungsweise die vorhandenen Tabellen entsprechend erweitert werden. Jede Tabelle muss über einen Schlüssel verfügen. In dem Beispiel sind dieses die „AW-Nr“ und die „S-Nr“. Über sie erfolgt die Datenzuordnung und die Verknüpfung der Daten untereinander, die so genannten Tabellen-Beziehungen. Ein Schlüssel kann auch aus mehreren

Einträgen bestehen. So ist es in der Tabelle „Einzugstermine“ der Fall. Näher erläutert werden die Funktionen der Schlüssel in Kapitel 5.4.5.

Das relationale Datenbanksystem hat aber auch Nachteile. So zum Beispiel ist die Suche nach Daten innerhalb der Datenbank aufwendiger. Das ergibt sich aus der Datensuche in mehreren Tabellen, bedingt durch die Normalisierung. Bei den hierarchischen Datenbanken muss jeweils nur eine Tabelle abgefragt werden. Bei den relationalen Datenbanken kann sich eine Abfrage über eine Vielzahl von Tabellen erstrecken. Diese komplizierteren Abfragen bleiben dem Anwender aber verborgen, da sie über die Programmierweise entsprechend hinterlegt werden können. Um dem Problem der aufwendigen Abfrage entgegen zu wirken, können so genannte Views über mehrere Tabellen gelegt werden, so dass die Datenbank die verschiedenen Tabellen verknüpft und nur eine Ergebnismenge zurückliefert. An dem Beispiel in Abbildung 13 betrachtet, würde das Ergebnis die Liste „Name“, „Straßenname“ und „Einzugstermin“ zurückliefern. Ein weiterer Nachteil für den Programmierer ist, dass die Datenbank mit jeder neuen Tabelle schwerer zu überschauen ist. Das Geflecht zwischen den Tabellen wird dadurch immer umfangreicher.

Zusammenfassend lässt sich nach [S09] sagen, dass sich relationale Datenbanken durch eine große Flexibilität hinsichtlich Änderungen und Ergänzungen an der Datenstruktur auszeichnen. Dadurch ist eine realitätsnahe Nachbildung möglich. Gerade die große Flexibilität macht sie jedoch schwer überschaubar. Daher ist eine gute Dokumentation der Datenstruktur und der verschiedenen Beziehungen zwischen den Tabellen von Nöten. Zur Vertiefung der Thematik zum ER-Modell ist das Buch [EN02] zu empfehlen. Dort wird auch auf Erweiterungen des ER-Modells eingegangen.

5.3 Datenbankbegriffe

Um sich in der Datenbankwelt zurecht zu finden, müssen einige Fachbegriffe geklärt sein. Dieses erfolgt am einfachsten anhand einer Tabelle, wie sie in Abbildung 14 zu finden ist.

Das Diagramm zeigt eine Tabelle mit dem Titel **Einzugstermine**. Die Spaltenüberschriften sind **AW-Nr**, **S-Nr** und **Einzugsdatum**. Die Datenzeilen sind:

| AW-Nr | S-Nr | Einzugsdatum |
|-------|------|--------------|
| 134 | 101 | 01.01.2003 |
| 257 | 101 | |
| 257 | 102 | 21.06.2008 |
| 313 | 110 | 18.09.2010 |
| 313 | 101 | 21.03.2012 |

Die Beschriftungen im Diagramm sind:

- Entität (Tabellenname)**: Ein Pfeil weist auf den Tabellentitel **Einzugstermine**.
- Attribut (Spaltenname)**: Ein Pfeil weist auf die Spaltenüberschriften **AW-Nr**, **S-Nr** und **Einzugsdatum**.
- Null-Wert**: Ein Pfeil weist auf den leeren Zellen in der Zeile mit AW-Nr 257 und S-Nr 101.
- Tupel (Datensatz)**: Ein Pfeil weist auf die gesamte Zeile mit AW-Nr 257 und S-Nr 101.
- Attributwert (Wert, Datum)**: Ein Pfeil weist auf den Wert **313** in der Spalte **AW-Nr**.

Abbildung 14: Datenbankbegriffe

Grundsätzlich ist bei den nachfolgenden Erklärungen eine Aufteilung der Bezeichnungen in das konzeptionelle²⁸ und das physische²⁹ Datenmodell zu finden. Letzteres ist nachfolgend in Klammern notiert. Sie sind in Teilen auch in der Abbildung 14 dargestellt. Zusammenfassend sind die wichtigsten Begriffe:

- **Entität (Tabellenname):** Themenkreis mit Elementen gleicher Merkmale
- **Entitätsmenge (Datensätze):** Summe der Werte der Tabelle, so genannte Tabelleninhalte
- **Tabelle:** Entität + Entitätsmenge
- **Tupel (Datensatz):** Entspricht einem vollständigen Datensatz, Wert eines Elements
- **Attribut (Spaltenname):** Merkmal eines Tupels
- **Attributwert (Wert, Datum):** Datenwert eines Attributes im Tupel
- **Null-Wert:** Es liegt für ein Attribut kein Attributwert vor. Es bedeutet nicht den Wert 0.
- **Domäne:** Wertebereich eines Attributes
- **Relation:** Beziehung zwischen Tabellen
- **Datenbasis:** Summe aller Tabellen
- **Datensystem:** Datenbasis + Applikationen + Zugriffsrechte
- **Datenbank:** Datenbasis + Datenverwaltungssystem

5.4 Beziehungen innerhalb einer Datenbank

Die Beziehung zwischen Tabellen innerhalb einer Datenbank beziehungsweise einer Domäne wird Assoziation genannt. Diese legt nach [S09] fest, wie viele Tupel aus einer ersten Tabelle mit einem Tupel aus einer zweiten Tabelle verbunden sind. Dabei gibt es vier verschiedene Assoziations-Typen. Diese sind in Tabelle 5 dargestellt und erklärt.

| Abkürzung | Assoziationstyp | Anzahl der Tupel, die in Tabelle 2 mit Tabelle 1 verbunden sind |
|-----------|------------------------------------|-----------------------------------------------------------------------------|
| 1 | einfache Assoziation | genau ein Tupel in Tabelle 2 ist verbunden mit Tabelle 1 (1) |
| c | konditionelle Assoziation | kein oder ein Tupel aus Tabelle 2 ist verbunden mit Tabelle 1 (0/1) |
| m | multiple Assoziation | mindestens ein Tupel aus Tabelle 2 ist verbunden mit Tabelle 1 (≥ 1) |
| mc | multiple-konditionelle Assoziation | beliebig viele Tupel in Tabelle 2 sind verbunden mit Tabelle 1 (≥ 0) |

Tabelle 5: Assoziationstypen

²⁸ Das konzeptionelle Datenmodell beschreibt nach [S09] die möglichen Beziehungen zwischen den Entitäten. Dabei finden die Beziehungen zwischen den Tabellen und die Attribute in diesen keine Berücksichtigung.

²⁹ Das physische Datenmodell beschreibt nach [S09] die Umsetzung des konzeptionellen Datenmodells auf der Datenebene. Über Primär- und Fremdschlüssel werden die Beziehungen zwischen den Tabellen hergestellt und aus den Entitäten werden Tabellen mit Attributen.

Jede Tabellenbeziehung enthält immer eine Assoziation und eine Gegenassoziation. So ist es möglich, die beiden Tabellen in Beziehung zueinander zu stellen. In den Kapiteln 5.4.1 bis 5.4.4 wird dieses exemplarisch für vier relevante Assoziations-Typen dargestellt. Zunächst sind in Tabelle 6 alle Assoziationsmöglichkeiten noch einmal aufgeführt. Sie lassen sich in hierarchische Beziehungen, konditionelle Beziehungen und netzwerkförmige Beziehungen aufteilen. Diese sind jeweils in Abhängigkeit zwischen zwei Tabellen dargestellt.

| Tabelle 1 \ Tabelle 2 | 1 | c | m | mc | |
|-----------------------|------|------|------|-------|-----------------------------|
| 1 | 1-1 | c-1 | m-1 | mc-1 | hierarchische Beziehungen |
| c | 1-c | c-c | m-c | mc-c | konditionelle Beziehungen |
| m | 1-m | c-m | m-m | mc-m | netzwerkförmige Beziehungen |
| mc | 1-mc | c-mc | m-mc | mc-mc | |

Tabelle 6: Beziehungstypen

Für relationale Datenbanksysteme, wie sie hier verwendet werden sollen, sind nur hierarchische Beziehungen (1-1, 1-c, 1-m, 1-mc) erlaubt. Alle anderen Beziehungen sind programmtechnisch nicht verarbeitbar und müssten transformiert³⁰ werden. Bei der Zusammenstellung der Datenbank sollte daher darauf geachtet werden, dass nicht erlaubte Beziehungstypen erst gar nicht entstehen. So bleibt eine Transformation erspart.

5.4.1 1-1 Beziehung

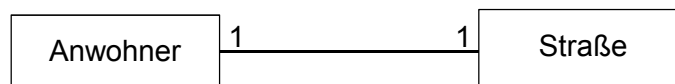


Abbildung 15: Entitätenblockdiagramm einer 1-1 Beziehung

In der Abbildung 15 ist eine 1-1 Beziehung zwischen zwei Datensätzen aufgezeigt. Die Darstellungsart nennt sich Entitätenblockdiagramm. Aus der Sicht des Anwohners gibt es genau eine Straße, die zu einem Anwohner gehört. Wenn die Sichtweise der Straße betrachtet wird, gibt es genau ein Anwohner, der zu dieser Straße gehört, also eine 1 zu 1 Beziehung.

Generell empfiehlt sich in so einem Fall eine Zusammenlegung der Tabellen, da immer nur ein Anwohner in einer Straße wohnen dürfte und kein zweiter Anwohner in der gleichen Straße wohnen kann. Erlaubt ist diese Zusammenlegung aber nur, wenn keine der zwei Tabellen in Beziehung zu einer anderen Tabelle steht. Sonst kann es Fehler in der Datenkonsistenz³¹ geben.

³⁰ Für die Transformation gibt es Umwandlungsregeln. Diese finden sich näher erläutert in [SSH10].

³¹ Daten in einer Datenbank sind konsistent, wenn sie logisch korrekt beziehungsweise widerspruchsfrei sind.

5.4.2 1-c Beziehung

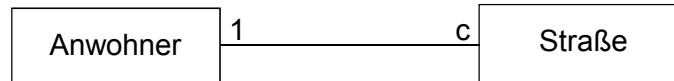


Abbildung 16: Entitätenblockdiagramm einer 1-c Beziehung

Die Abbildung 16 zeigt eine 1-c Beziehung. Aus Sicht der Straße kann genau ein Anwohner pro Straße wohnen. Aus der Gegenseite des Anwohners betrachtet kann dieser in einer Straße wohnen oder nicht. Zusammenfassend ist also eine Straße durch genau einen Anwohner bewohnt oder nicht. Bei diesem und auch den nachfolgenden Beziehungstypen ist die Verknüpfung der Tabellen über einen Schlüssel erforderlich. Näheres dazu folgt im Kapitel 5.4.5.

5.4.3 1-m Beziehung

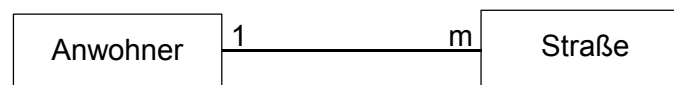


Abbildung 17: Entitätenblockdiagramm einer 1-m Beziehung

Im Entitätenblockdiagramm der Abbildung 17 ist eine 1-m Beziehung ersichtlich. Aus der Sichtweise der Straße kann diese durch genau einen Anwohner bewohnt werden. Wo hingegen ein Anwohner in mindestens einer Straße wohnt. Es wäre also möglich, dass der gleiche Anwohner mehrere Straßen bewohnt (also mehrere Häuser hat), mindestens jedoch ein Haus in einer Straße. Zu beachten ist, dass auch hier immer genau ein Anwohner je Straße wohnt.

5.4.4 1-mc Beziehung

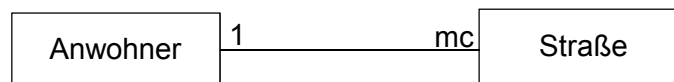


Abbildung 18: Entitätenblockdiagramm einer 1-mc Beziehung

Bei der 1-mc Beziehung, wie sie in Abbildung 18 dargestellt ist, kann wieder in einer Straße genau ein Anwohner wohnen. Jeder Anwohner kann aber in beliebig vielen Straßen wohnen. Es kann also der gleiche Anwohner in so vielen Straßen wohnen (mit je einem Haus) wie er möchte oder auch in keiner Straße wohnen. Die 1-mc Beziehung stellt eine Kombination der 1-c Beziehung und der 1-m Beziehung dar. Der Unterschied zur 1-m Beziehung ist hier, dass der Anwohner auch in keiner Straße wohnen kann.

5.4.5 ID-Schlüssel und Primär-/Fremdschlüssel

Eine minimale Menge von Attributen, deren Werte den zugeordneten Tupel eindeutig innerhalb aller Tupel eines Typs identifizieren, nennt man nach [KE11] Schlüssel. Um die Beziehungen zwischen Tabellen darstellen zu können, werden im Folgenden die Begriffe Identifikationsschlüssel, Primärschlüssel und Fremdschlüssel erklärt.

Der **Identifikationsschlüssel** (ID-Schlüssel) dient der Identifizierung eines Tupels innerhalb einer Tabelle. Bei einem Datenbanksystem ist es wichtig, dass jeder Tabelleneintrag

eine Eindeutigkeit besitzt. Anhand dieses Schlüssels werden die Datensätze identifiziert und zugeordnet. Der Wert des Identifikationsschlüssels darf kein Null-Wert sein. Dieser Wert besitzt keine Eindeutigkeit. Gebildet wird dieser Schlüssel aus einem Attribut oder einer Attributkombination eines Tupels. Dabei ist darauf zu achten, dass es keine Doppelungen gibt. Beispielsweise würde bei der Tabelle „Anwohner“ der Name nicht als Identifikationsschlüssel geeignet sein, da Namen doppelt vorkommen könnten. In diesem Fall kann eine Attributkombination, etwa aus Name und Vorname des Anwohners gewählt werden. Dabei muss aber auch sichergestellt sein, dass diese Kombination nicht doppelt vorkommt. Eine weitere Möglichkeit ist es, ein neues Attribut einzufügen. Beispielsweise ein so genannter Surrogatekey³², wobei es sich um eine fortlaufende Nummer handelt. Dabei ist sichergestellt, dass dieser nicht mehrfach vorkommen kann.

Eine Tabelle lässt sich auch in Textform darstellen. Nimmt man die Tabelle „Anwohner“ aus Abbildung 13, so enthält sie die Attribute „AW-Nr“ und „Name“. Nach [H97] wird bei einer textuellen Darstellung der Identifikationsschlüssel unterstrichen. Die einzelnen Attribute einer Tabelle werden innerhalb einer Klammer immer durch Kommata getrennt. Wenn die „AW-Nr“ der ID-Schlüssel ist, sieht die textuelle Darstellung wie folgt aus:

Anwohner (AW-Nr, Name)

Die Zuordnungen zwischen den Tabellen erfolgen über **Primärschlüssel** und **Fremdschlüssel**. Über sie werden die Beziehungen hergestellt. Unterschieden werden muss zunächst zwischen Stammdaten (Mastertabelle) und Bewegungsdaten (Slavetabelle). In der Abbildung 19 sind drei Tabellen dargestellt.

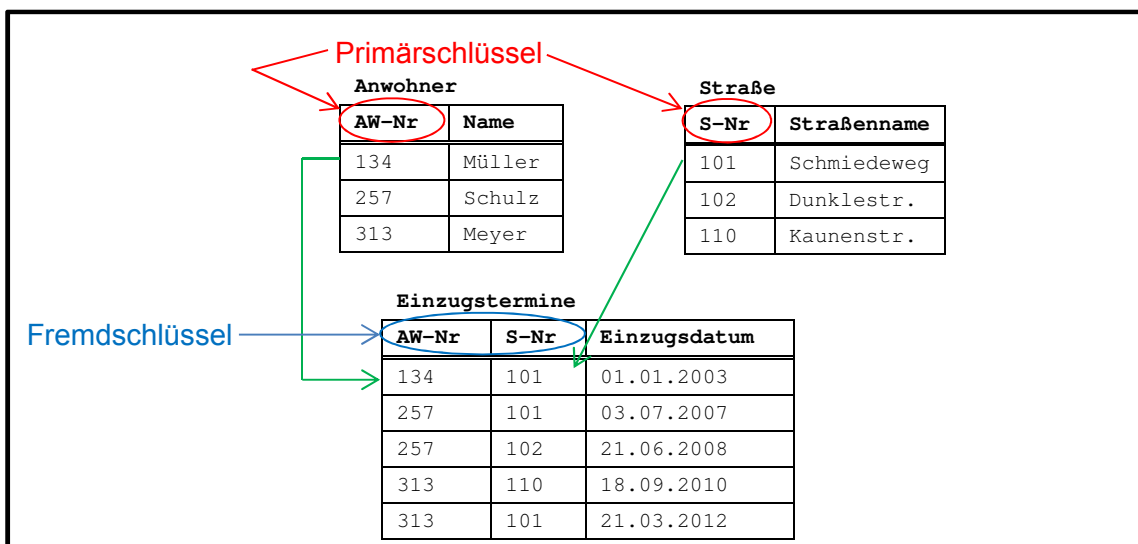


Abbildung 19: Primär- und Fremdschlüssel für Tabellenbeziehungen

Die Stammdaten stehen in den Tabellen „Anwohner“ und „Straße“, die Bewegungsdaten in der Tabelle „Einzugstermine“. Als Stammdaten sind die Daten zu betrachten, die einen gewissen Ursprungscharakter innerhalb der Datenbank bilden. Die Bewegungsdaten wie-

³² Ein Surrogatekey ist ein künstlicher Schlüssel, der selbst keine Informationen trägt.

derum entstehen aus den Stammdaten. Es handelt sich also um Daten, welche sich ständig verändern. So zum Beispiel bei einem Umzug eines Anwohners in eine andere Straße.

Der Primärschlüssel einer Mastertabelle wird häufig aus dem Identifikationsschlüssel gebildet. Der Fremdschlüssel in einer Slavetabelle setzt sich aus dem Primärschlüssel einer oder mehrerer Mastertabellen zusammen. So bilden die jeweiligen Primärschlüssel der beiden Mastertabellen in Abbildung 19 den Fremdschlüssel der Slavetabelle. Zu beachten ist, dass der Wertebereich der Primärschlüssel mit dem Wertebereich des Fremdschlüssels übereinstimmt. Es muss sich also um die gleiche Domäne handeln. Daher ist der Fremdschlüssel in der Slavetabelle auch aus zwei Attributen zusammengesetzt. Ist dieses nicht der Fall, ist eine eindeutige Zuordnung der Tupel zwischen den Tabellen nicht möglich. In einigen Datenbankprogrammen wird für die Begriffe Identifikationsschlüssel, Primärschlüssel und Fremdschlüssel nur der Begriff Primärschlüssel verwendet.

5.5 Normalisierungsprozess

Der Normalisierungsprozess wird unter anderem für die Erkennung redundanter Daten in einer Datenbank genutzt. Eine redundanzfreie Speicherung dient der Einsparung von Speicherplatz, da die Ablage von doppelten Daten vermieden wird. Des Weiteren wird durch den Normalisierungsprozess Anomalien und Fehlern vorgebeugt, die durch die vielen Tabellen entstehen könnten. Grundsätzlich ist nach [S09] eine redundanzfreie Speicherung der Daten gegeben, wenn kein Teil des Datenbestandes weggelassen werden kann, ohne dass ein Informationsverlust entsteht.

Sogenannte Mutationsanomalien können entstehen, wenn es zu mehrfacher Datenhaltung (Redundanzen) kommt. Wenn beispielsweise der Anwohner Müller im Schmiedeweg und der Dunklestr. wohnt und die Anwohnernummer nur bei einem Tabelleneintrag geändert wird, gibt es auf einmal zwei verschiedene Müllers. Eigentlich sind es aber die gleichen, die nur in zwei verschiedenen Straßen wohnen. Daher gilt es, die Tabellen so zu strukturieren, dass solche Fehler nicht existieren können. In diesem Beispiel empfiehlt sich eine Aufteilung in mehrere Tabellen.

Im Folgenden wird zunächst der Begriff Abhängigkeit innerhalb einer Datenbank erklärt. Dieses dient dem Verständnis des Normalisierungsprozesses und der Normalformen.

| Attribut A | B | B | B | C |
|------------|----|----|----|------|
| 010201 | 01 | 02 | 01 | 0916 |
| 010304 | 01 | 03 | 04 | 1114 |

Abbildung 20: Übersicht der Abhängigkeiten

Bei den Abhängigkeiten gibt es drei Typen. Diese sind funktionale, volle und transitive Abhängigkeiten. Dargestellt sind sie in Abbildung 20.

Nach [P11] lassen sich funktionale Abhängigkeiten wie folgt erklären: Ein Attribut (oder eine Attributkombination) B ist von einem Attribut (oder einer Attributkombination) A funktional abhängig, wenn zu einem Attributwert von A genau ein Attributwert von B gehört. Es ergibt sich also aus dem Attributwert A eindeutig der Attributwert B. Ersichtlich ist diese Abhängigkeit in der Abbildung 20 an den blauen Pfeilen.

Die volle Abhängigkeit ergibt sich, wenn ein Attribut C von B abhängig ist, nicht jedoch bereits von einem Teil der Attributkombination B funktional abhängig ist. Demnach darf C nicht aus B zusammengesetzt sein aber dennoch in Zusammenhang mit B stehen. Dargestellt ist der Zusammenhang mit den grünen Pfeilen in der vorhergehenden Abbildung.

Nach [S09] sind transitive Abhängigkeiten folgendermaßen zu verstehen: Ein Attribut C ist vom Attribut A transitiv abhängig, wenn das Attribut C vom Attribut B abhängig ist und das Attribut B von A. Im diesem Fall würde eine Dreiecks-Abhängigkeit bestehen. Diese Art der Abhängigkeit gilt es zu vermeiden. Die orangenen Pfeile in der Abbildung 20 stellen diese Abhängigkeit noch einmal dar.

Der Normalisierungsprozess erfolgt bei der Tabellenstrukturierung anhand von sogenannten Normalformen (NF). Eine Datenbank gilt ab der dritten Normalform als normalisiert. Die bis dahin in den Tabellen enthaltenen Daten sind redundanzfrei. Dieses gilt an der Stelle aber noch nicht für den gesamten Datenbestand innerhalb einer Datenbank. Zusammenfassend verstehen sich nach [P07] die ersten drei Normalformen wie folgt:

- 1. NF: Eine Tabelle befindet sich in der ersten Normalform, wenn alle Attribute einfache Attributwerte aufweisen (Null-Wert ist erlaubt).
- 2. NF: Eine Tabelle befindet sich in der zweiten Normalform, wenn alle Attribute, die nicht zum Identifikationsschlüssel gehören, voll funktional von diesem abhängig sind.
- 3. NF: Eine Tabelle befindet sich in der dritten Normalform, wenn keine transitiven Abhängigkeiten vorliegen.

Basierend auf den ersten drei Normalformen gibt es zwei weitere Normalformen. Diese dienen der Normalisierung des gesamten Datenbestandes. Die Anwendung der vierten und fünften Normalform ist aber nicht zwingend bei der Datenbankerstellung.

Eine Datenbasis, die auf der vierten Normalform beruht, wird als redundanzfrei betrachtet. In diesem Fall gilt das für die gesamte Datenbank, also den kompletten Datenbestand. Das Erreichen und Einhalten der fünften Normalform stellt den Idealzustand dar. Dieses ist aber in der Praxis oft nicht möglich.

Zusammenfassen lauten die beiden Normalformen:

- 4. NF: Eine Datenbasis befindet sich in der vierten Normalform, wenn alle Tabellen in der 3. Normalform sind und nur lokale³³ und globale³⁴ Attribute vorkommen, also keine Mehrfachverwendung in der gleichen Tabelle vorliegt.
- 5. NF: Die Domänen und Wertebereiche der globalen Attribute müssen übereinstimmen (in Master- und Slavetabelle).

5.6 Strukturregeln

Die Überprüfung der Datenbasis findet anhand von sechs Strukturregeln statt. Überprüft werden dabei die Tabellen der Datenbankanwendung. Wenn alle Tabellen der Datenbasis diesen Regeln entsprechen, gilt die Datenbasis als global normalisiert. Zum Verständnis der Strukturregeln werden zuvor noch die Begriffe statischer- und dynamischer Wertebereich erklärt.

Ein statischer Wertebereich bezieht sich auf die Werte eines Attributes. So kann beispielsweise das Attribut Straße auf eine gewisse Straße beschränkt oder das Einzugsjahr auf Werte zwischen 1900 und 2200 begrenzt werden. Demzufolge wird nur eine begrenzte Menge an Werten zugelassen. Ein dynamischer Wertebereich ist eine Menge von Schlüsselwerten oder deren Kombinationen, die für einen Fremdschlüssel zur Verfügung stehen. Der dynamische Wertebereich kommt also bei Tabellenbeziehungen zum Einsatz. Wenn ein Schlüsselwert in Tabelle A einen festen (statischen) Wertebereich hat und dieser mit Tabelle B verknüpft wird, so hat er in Tabelle B einen dynamischen Wertebereich, da er sich in Tabelle B undefiniert verändert.

Nachfolgend sind die Strukturregeln (SR) noch einmal dargestellt, diese sind nach [S09]:

- SR 1: Jede Tabelle muss einen Identifikationsschlüssel haben.
- SR 2: Die Datenbasis sollte aus Tabellen der dritten Normalform bestehen und nur lokale und globale Attribute dürfen existieren (vierte Normalform).
- SR 3: Lokale Attribute müssen statische Wertebereiche aufweisen.
- SR 4: Rekursive³⁵ Tabellenbeziehungen sind verboten.
- SR 5: Bei Unter- und Obermengen-³⁶ Beziehungen muss eine Spezialisierung³⁷ mit zugelassener³⁸ Überlappung vorliegen.

³³ Lokale Attribute sind alle Attribute einer Tabelle, die nicht zum Identifikationsschlüssel gehören.

³⁴ Globale Attribute müssen mindestens in einer Tabelle vorkommen und Teil des Identifikationsschlüssels sein.

³⁵ Rekursive Tabellenbeziehungen beziehen sich wieder auf sich selbst.

³⁶ Unter- und Obermengen entstehen bei einer Aufteilung von Attributen, wenn beispielsweise das Attribut „Straße“ in der Tabelle A, bei der Tabelle B in „Straßenname“ und „Hausnummer“ aufgeteilt wird. „Straße“ ist die Obermenge und „Straßenname“ + „Hausnummer“ die Untermenge. Wenn diese Attribute in Beziehung stehen, muss eine Spezialisierung erfolgen.

- SR 6: Bei Einführung globaler Attribute in Tabellen als Fremdschlüssel ist deren Wertebereich einzuhalten (Einschränkung des dynamischen Wertebereichs).

Die Strukturregeln sind für die Erstellung und spätere Datenhaltung in einer Datenbank unabdingbar. Sollten sie nicht eingehalten werden, kann es zu Fehlfunktionen in der Datenbank kommen.

³⁷ Spezialisierung steht für die Zusammenfassung von Tabelle A und Tabelle B in Tabelle C.

³⁸ Von zugelassener Überlappung wird gesprochen, wenn in Tabelle A die Attribute von Tabelle B und Tabelle C enthalten sind.

6 Softwareentwurf

6.1 Strukturfestlegung

Die Strukturfestlegung dient der Ermittlung der Daten, die für den Aufbau und die Funktionalität der Datenbank benötigt werden. Die Inhalte wurden per Interview ermittelt. Dabei wurden die später nutzenden Abteilungen befragt, welche Inhalte im Detail an welcher Stelle erfasst werden müssen. Um möglichst kosteneffizient zu arbeiten, ist eine gewissenhafte Abarbeitung dieser Schritte notwendig. Laut [KE11] lässt sich abschätzen, dass ein Fehler während der Anforderungsanalyse noch mit Kosten von 1 Euro zu korrigieren ist. In der Entwurfsphase verursacht dieser bereits 10 Euro Kosten und in der Realisierungsphase verursacht die Korrektur Kosten in Höhe von 100 Euro.

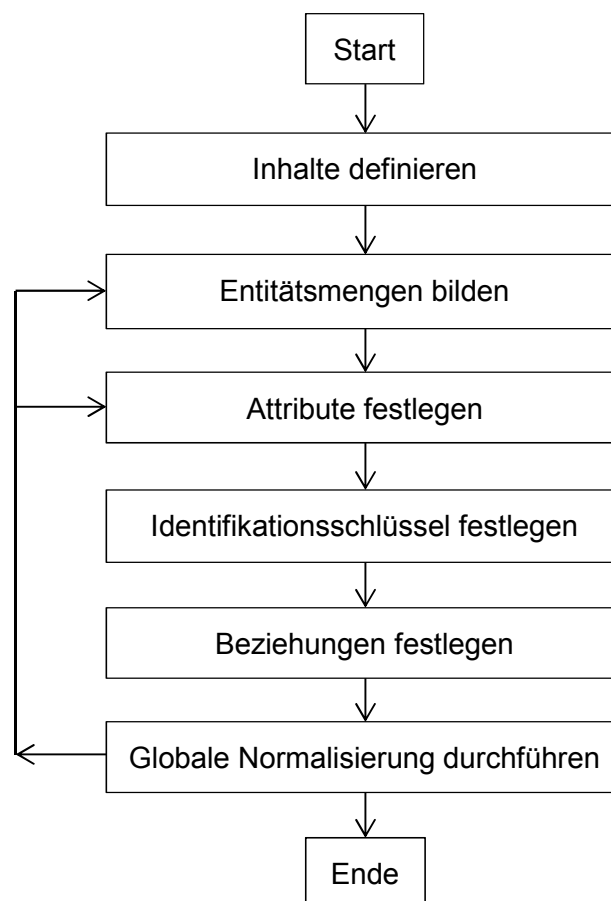


Abbildung 21: Ablaufdiagramm für den Entwurfsprozess

Die Abbildung 21 zeigt nach [EN09] den Ablauf des Entwurfsprozesses. Dabei werden als Erstes die Inhalte definiert, die in der Datenbank erfasst werden sollen. Dazu erfolgt eine Ergänzung der Daten aus Kapitel 2 und 3 mit den Daten aus oben erwähntem Interview. Als nächster Schritt wird die Festlegung der Entitäten vorgenommen, das bedeutet die Daten werden zu logischen Tabellen zusammengefasst. Es wird eine Mastertabelle erstellt. Die Slavetabellen beinhalten die Zulieferdaten für die Mastertabelle. Im nächsten

Schritt werden die Attribute für die Tabellen festgelegt. Es werden also die Inhalte, welche später in den Tabellen stehen sollen, aufgeteilt. Da an dieser Stelle alle Inhalte festgelegt sind, kann ein Identifikationsschlüssel für jede Tabelle erstellt werden. Diesen benötigt jede Tabelle innerhalb der Datenbank.

Als nächstes müssen die Beziehungen zwischen den Tabellen hergestellt werden. Ohne sie ist eine Verknüpfung der Tabelleninhalte nicht möglich, da sonst kein Zusammenhang zwischen den einzelnen Daten hergestellt werden kann. Den letzten Schritt beim Entwurfsprozess stellt die globale Normalisierung dar. Dabei werden alle Inhalte gemäß den Normalisierungsregeln von Kapitel 5.5 durchleuchtet und gegebenenfalls korrigiert. Dieses ist auch an den Pfeilverläufen in Abbildung 21 zu sehen. Während des gesamten Entwurfsprozesses wird sich immer an die Strukturregeln aus Kapitel 5.6 gehalten, damit eine lauffähige Datenbankanwendung garantiert werden kann [EGH⁺92].

6.2 Tabellenzusammenstellung

Wie in Kapitel 6.1 zu sehen ist, folgt als Erstes die Entitäten-Festlegung anhand der Inhalte aus den Kapiteln 2 und 3. Demnach lassen sich folgende Entitäten bilden:

- Trafodaten
- Standort Anlagen
- Standort Trafokammern
- Eigentum
- Wartung
- Spannungen
- Stellungen primär
- Stellungen sekundär
- Ölproben selbst
- Ölproben fremd
- Isolierung
- Status

Diese Entitäten ergeben sich teils aus den Inhalten und teils aus logischen Überlegungen zum Aufbau der Datenbank. Die Tabelle „Trafodaten“ wird die Mastertabelle in der Anwendung bilden. Bei allen anderen Tabellen handelt es sich um Slavetabellen. Diese werden später in Beziehung zur Mastertabelle stehen. Damit wird den Empfehlungen für den Datenbankentwurf aus [KE11] gefolgt.

Die Slavetabellen sind jeweils in Gruppen aufgeteilt. So ist eine Aufgliederung der einzelnen Themengebiete möglich. Des Weiteren beugt diese Aufteilung in der Entstehungsphase auch Redundanzen vor. Wenn die Themengebiete vermischt auftreten, birgt das ein höheres Risiko für Dopplungen bei der Datenerfassung.

Die Tabellen der Standorte unterteilen sich in die Anlagen- und Trafokammer-Erfassung. Diese Unterscheidung ist nötig, da Trafokammern mit den gleichen Merkmalen in ver-

schiedenen Anlagen vorkommen können. Gleichzeitig stehen immer mehrere Transformatoren an einem Standort. Daher dient die Tabelle „Standort-Anlagen“ auch als Zuliefertabelle für die Tabelle „Trafodaten“. So wird auch hier Inkonsistenzen vorgebeugt. Wenn beispielsweise ein Standortname sich ändert, müssten alle einzelnen Trafo-Datensätze angepasst werden. Das Risiko einen Datensatz dabei zu vergessen, ist sehr hoch. Daher wird das Attribut „Standort“ in einer separaten Tabelle ausgelagert.

In den Tabellen „Eigentum“ und „Wartung“ werden Daten der jeweiligen Betreuungsbereiche bereitgestellt. Auch hier ist eine Unterscheidung nötig. Die Bereiche und Eigentumsverhältnisse stimmen nicht in Gänze überein. Da diese Daten später als Drop-Down-Liste zur Verfügung stehen sollen, wären auch falsche Einträge enthalten, wenn diese Daten in einer Tabelle geführt werden würden. Daher erfolgt die Aufteilung in zwei getrennte Tabellen mit ähnlichen Daten. In den erwähnten Drop-Down-Listen steht jeweils der dynamische Wertebereich (nach Kapitel 5.6) der beiden Slavetabellen als Auswahlmöglichkeit zur Verfügung.

Eine weitere Tabelle ist die der „Spannungen“. Jeder Transformator, der später in der Datenbank erfasst wird, wird einer Spannungsgruppe zugeordnet. Diese Maßnahme dient der schnellen Filterung bei Suchfunktionen und der Übersicht für die Benutzer bei der Datensatzauswahl. Die Angaben, die später an dieser Stelle enthalten sein werden, sind beispielsweise 30/6kV oder 6/0,5kV. Es ist immer eine Angabe über die Oberspannungs-³⁹ und die Unterspannungsseite⁴⁰ enthalten. Damit wird der Transformator eingruppiert. Um Abweichende Angaben an dieser Stelle zu vermeiden, wird diese Auswahl später ebenfalls als Drop-Down-Liste zur Verfügung stehen.

Die Tabellen der „Stellungen“⁴¹ sind auch in primär und sekundär unterteilt. Die Existenz dieser beiden Tabellen ist erforderlich, da ein Transformator mehrere Stellungen haben kann. Die meisten der eingesetzten Transformatoren haben fünf Stellungen, es sind am Standort Salzgitter aber auch Transformatoren mit bis zu 19 Stellungen verbaut. Wenn in der Mastertabelle für alle 19 Einträge immer Platz gelassen werden würde, wären beim überwiegenden Teil der Transformatoren Null-Werte vorhanden. So würde wiederum Speicherplatz verschenkt werden. Grundsätzlich gibt es bei einem Transformator mehrere Stellungen an der Oberspannungsseite. Die Unterspannungsseite ist immer gleich. Diese Technik rührt aus dem geringeren Stromfluss auf der Primärseite. Da der Aufbau in der Datenbank aber symmetrisch sein soll, wird sich auch für eine eigene Tabelle für die Sekundärseite entschieden.

³⁹ Oberspannungsseite ist die Bezeichnung für die Primärseite eines Transformators, also der Spannungseingang bei normaler Verwendung des Transformators. Die Angaben erfolgen in Volt (V) beziehungsweise Kilovolt (kV).

⁴⁰ Unterspannungsseite bezeichnet die Sekundärseite eines Transformators, also den Spannungsausgang bei normaler Verwendung des Transformators. Die Angaben erfolgen in Volt (V) beziehungsweise Kilovolt (kV).

⁴¹ Bei den Stellungen handelt es sich um Anzapfungen an den Wicklungen der Trafospulen. Über eine Veränderung der Stellung werden Schwankungen im Spannungsnetz ausgeglichen. Diese können durch unterschiedlich starke Belastungen des Transformators entstehen.

Zwei weitere Tabellen dienen der Erfassung der Ölproben. Eine Unterteilung erfolgt an dieser Stelle in eigene Ölproben und fremde Ölproben. Eigene Ölproben werden im werk-eigenen Labor erstellt und fremde stammen aus externen Laboren. Da es theoretisch im Lebenszyklus eines Transformators unendlich viele Ölproben geben kann, erfolgt die Unterbringung dieser Daten in separaten Tabellen. Diese werden später über Beziehungen in Verbindung mit den richtigen Datensätzen gebracht. In der Praxis erfolgt die Analyse einer Ölprobe in Abhängigkeit des Zustands und Alters des Transformators. Das heißt, dass alle drei bis fünf Jahre eine Ölprobe analysiert wird. Bei einer Lebensdauer pro Transformator von circa 60-70 Jahren sind eine entsprechende Anzahl von Probenberichten zu erwarten. Diese Daten werden für den Benutzer der Datenbank zusätzlich visuell aufbereitet. Näheres dazu folgt im Kapitel 6.6.

Die zwei letzten Entitäten von der Auflistung am Anfang des Kapitels sind „Isolierung“ und „Status“. Die Unterbringung erfolgt auch hier in separaten Tabellen, da diese als Drop-Down-Liste zur Verfügung stehen sollen. Diese Maßnahme dient auch hier dem Vorbeugen von Falscheinträgen. Bei der Isolierung werden die Möglichkeiten „Öl-Trafo“⁴² und „Gießharz-Trafo“⁴³ zur Verfügung gestellt. Beim Status wird zwischen „Installiert“, „Reserviert“ und „Verschrottet“ unterschieden. Die drei Kriterien besagen, ob es sich um einen angeschlossenen Transformator, einen in Reserve befindlichen Transformator oder um einen verschrotteten Transformator handelt. Auch von bereits entsorgten Transformatoren werden die Daten für Vergleich- und Statistik-Zwecke noch weiterhin gespeichert.

An dieser Stelle sind alle Entitätsmengen gebildet und unterteilt. Im nächsten Schritt werden die Attribute zu den Entitäten geordnet. Parallel dazu werden auch die Identifikationsschlüssel für die Tabellen festgelegt. Die Attributfestlegung erfolgte auch an dieser Stelle per Interview. Dabei wurden die betroffenen Abteilungen des programmeinsetzenden Unternehmens erneut befragt. Alle Daten, die für die Verwaltung von Transformatoren benötigt werden, wurden auf diese Weise erfasst.

Die Zuordnung der Attribute zu den einzelnen Entitäten erfolgt dann in einem separaten Schritt. Diese Entscheidungen werden beim Datenbankentwurf auf einer logischen Ebene getroffen. Bei der Zusammenstellung der Attribute ist äußerste Sorgfalt zu walten. Ein späteres Hinzufügen oder Umstrukturieren der Attribute zieht vielfach Arbeit nach sich, da alle Daten bis auf die Grundzusammensetzung der Datenbank eventuell angepasst werden müssten. Auf Basis dieser Befragung ergibt sich nachfolgende Tabellenzusammensetzung für die Mastertabelle:

⁴² Öl-Trafo bezeichnet einen Transformator, welcher als Kühl- und Isoliermedium Isolier-Öl um den Trafokern hat. Diese Transformatoren haben eine Metall Außenhülle und zwischen dieser und den Wicklungen (Kern) befindet sich Öl.

⁴³ Gießharz-Trafos haben einen Metall-Kern, welcher mit einem Kunststoff (Gießharz) überzogen ist. Das Isolier- und Kühlmedium stellt in diesem Fall das Gießharz und die Umgebungsluft dar. Die Wicklungen sind also mit entsprechendem Abstand zueinander montiert.

- Trafodaten (Laufende-Nr, Standort, Kammer-Nr, Hersteller, Typ, FabrikNummer, BauJahr, BauMonat, Isolierung, Status, Eigentum, Wartung, Umspanner, Ident-Spannungen, Bemerkung, NennLeistung, Schaltgruppe, Leerlauf-Verluste, Kurzschluss-Verluste, Schaltbild, Technische-Daten, Bemerkung-Elektrik, Laenge, Breite, Hoehe, RollenAbstand-Laengsfahrt, RollenAbstand-Querfahrt, GesamtGewicht, OelGewicht, OelVolumen, Zeichnung, Verladeplan, Bemerkung-Mechanik)

Für die Slavetabellen ergeben sich folgende Attribute:

- Standort-Anlagen (StandortAnlagen, AnlagenNummer, EigentumAnlagen, ZeichnungAnlagen, BemerkungAnlagen)
- Standort-Kammern (Kammer-Nr, StandortAnlagen, Kammer, AnlagenNummer, Max-Belastung, Statik-Vorhanden, Auffangvolumen, EigentumAnlagen, ZeichnungKammer, BemerkungKammer, Belegt)
- Eigentum (Eigentum)
- Wartung (Wartung)
- Spannungen (Spannungs-Nr, Spannung)
- Stellung-Primaer (Stellung-Nr, Laufende-Nr, Stellung, Spannung, Strom, UK, NennWert)
- Stellung-Sekundaer (Stellung-Nr, Laufende-Nr, Stellung, Spannung, Strom, NennWert)
- Oelproben-selbst (Proben-Nr, Laufende-Nr, Datum, Tan-Delta, Durchschlags-spannung, PCB-Vorhanden, PCB-Hoehe, LaborProbenNr, Neutralisationszahl, Verseifungszahl, GasImOel, Furananalyse, NaechsteProbe, Probenbericht, Bemerkung)
- Oelproben-fremd (Proben-Fremd-Nr, Laufende-Nr, Datum, Firma, Entspricht-VDE, NaechsteProbe, ProbenberichtFremd, Bemerkung, ProbeAuffaellig)
- Isolierung (Isolierung)
- Status (Status-Nr, Status)

Darüber hinaus gibt es eine weitere Slavetabelle. Diese wird nicht in Beziehung zu den anderen Tabellen stehen und in Kapitel 6.6 nähere Bedeutungen finden.

- Grenzwerte (Grenzwerte-Nr, NennSpannung, DurchschlagSpannung, Tan-Delta, DurchschlagSpannung-Gefahr, Tan-Delta-Gefahr, MaxJahreOelprobe)

Wie im Kapitel 5 erklärt wird, ist es unerlässlich für jede Tabelle einen Primärschlüssel festzulegen. In den vorhergehenden Auflistungen der Tabellen ist dieser bereits an dem unterstrichenen Attribut zu erkennen. Anhand dieses Schlüssels werden die Datensätze innerhalb der Datenbankanwendung verwaltet und zugeordnet.

Das Erstellen einer Tabelle ist bei der Datenbankanwendung Access ebenfalls der erste Schritt bei der Entwicklung. Auf diesen Tabellen basiert die gesamte Datenbank mit all ihren Funktionen. In der Abbildung 22 ist der Menüpunkt „Erstellen“ von Access 2010 zu sehen. Mit Hilfe dieser Menüleiste werden neue Tabellen angefertigt. Auch Formulare,

Abfragen und Berichte können über diese Leiste hinzugefügt werden. Die Funktionserläuterungen dazu folgenden in den nächsten Kapiteln. Die Makrofunktionen beinhalten auch Visual Basic. Darauf wird im Kapitel 6.6 stellenweise eingegangen.

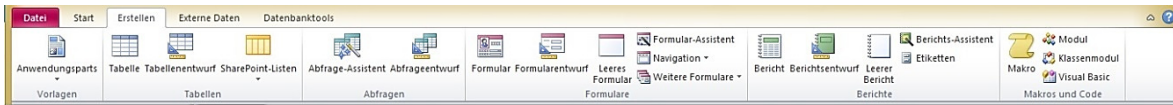


Abbildung 22: Menüleiste "Erstellen" in Access 2010

Über den Button „Tabelle“ in der Abbildung 22 gelangt man in die nachfolgende Ansicht der Datenbankoberfläche, wie sie in Abbildung 23 zu sehen ist. Über den Button „Ansicht“ (links oben in der Abbildung) kann zwischen verschiedenen Ansichten auf die Oberfläche umgestellt werden. So ist es auch bei anderen Funktionen möglich, Objekte im Entwurf zu betrachten oder die endgültige Ansicht zu sehen, so wie sie der spätere Nutzer zu Gesicht bekommt.

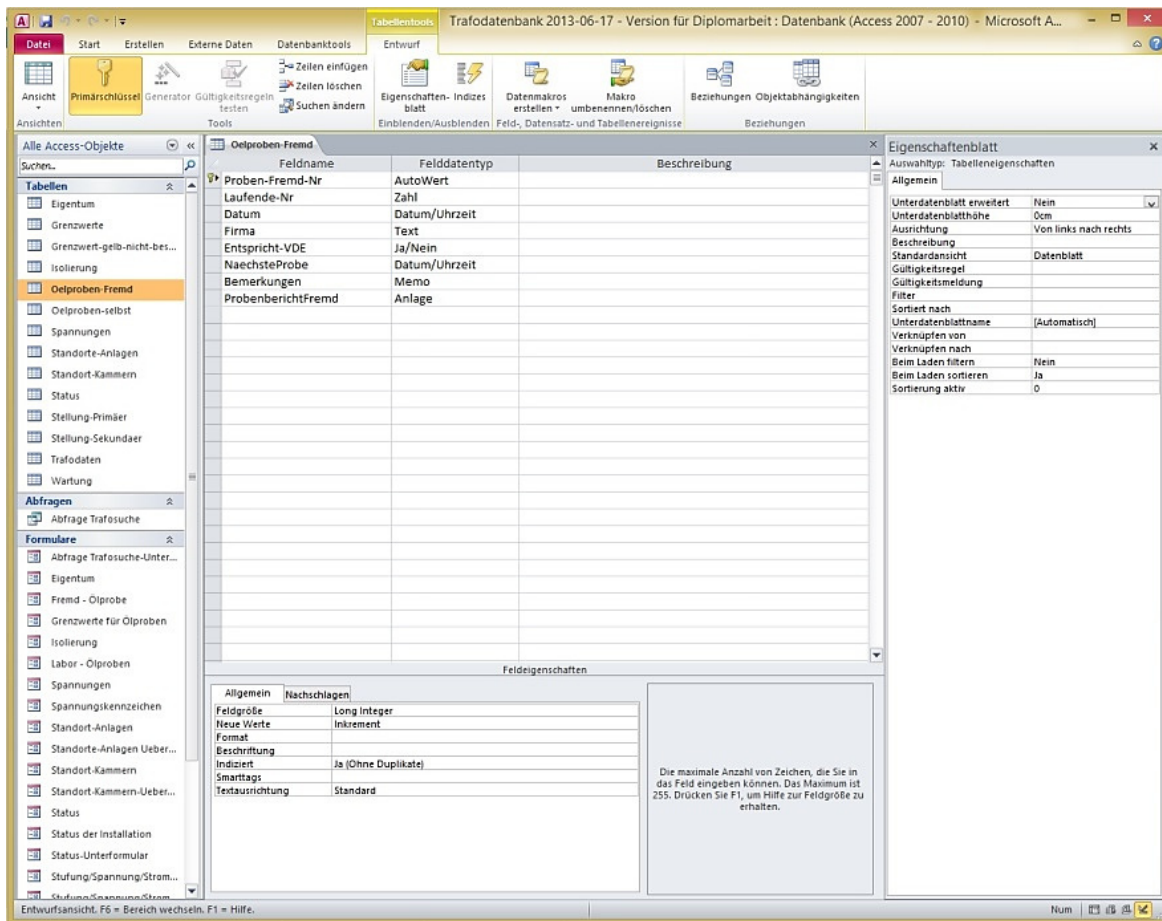


Abbildung 23: Tabellenerstellung unter Access 2010

In der sogenannten Entwurfsansicht, wie sie in Abbildung 23 dargestellt ist, können alle Attribute einer Tabelle eingetragen werden. Dieses erfolgt in der Spalte „Feldname“. Unter „Felddatentyp“ können verschiedene Eigenschaften des Attributes festgelegt werden. Bei-

spielsweise sind dies: Zahl, Text, Auswahl (Ja/Nein), Datum/Uhrzeit, Memo⁴⁴ oder Anlage⁴⁵. Im unteren Bereich der Feldeigenschaften können die Details für den jeweiligen Attributtyp eingestellt werden. Wie in der Abbildung 23 ebenfalls zu sehen ist, wird das Attribut „Proben-Fremd-Nr“ als Primärschlüssel gewählt. Daher steht beim Felddatentyp auch „AutoWert“ in der Auswahl. Die Auswahl des Primärschlüssels selbst erfolgt über den entsprechenden Button (oben in der Abbildung zu sehen). Felder, die als Primärschlüssel genutzt werden sollen, müssen indiziert sein. Das bedeutet, dass unter anderem eine zusätzliche Überwachung aktiv ist, die keine mehrfach gleichen Einträge in dieses Feld zulassen wird. Das geschieht implizit durch die Kennzeichnung des Feldes als Primärschlüssel.

Bei einer Ansichtsumschaltung in die so genannte Datenblattansicht sieht die Tabelle aus wie bei einer herkömmlichen Tabellen-Anwendung. Hier könnten bereits einzelne Tupel eingegeben werden. Dieses empfiehlt sich aber nicht. Daher sollte diese Funktion nur zur Kontrolle beim Entwurfsprozess oder zur späteren Wartung genutzt werden. Die Daten einer Datenbank Anwendung sollen immer über die spätere Benutzeroberfläche geändert werden. Sonst kann es zu Fehlfunktionen innerhalb der Datenbank kommen. Auch können Daten falsch eingruppiert und dadurch verloren gehen. So oder so sollen die Daten aber erst nach Fertigstellung der Datenbank eingegeben werden.

Wie eben exemplarisch für die Tabelle „Ölproben-Fremd“ beschrieben, werden auch alle anderen Tabellen für die Datenbank erstellt. Zu sehen sind sie im linken Bereich der Auflistung, in der Abbildung 23. Diese ist nach den Gruppen Tabellen, Abfragen und Formulare sortiert. Auch die Berichte stehen nach ihrer Erstellung mit in dieser Auflistung. Alle fertiggestellten Elemente einer Gruppe sind an der Stelle abgebildet.

6.3 Tabellenbeziehungen

Nachdem alle Tabellen erstellt sind, folgt der nächste Schritt. Die Beziehungen zwischen den Tabellen müssen hergestellt werden. Dieses wird unter Access mit der Funktion „Beziehung“ bewerkstelligt. Zu finden ist sie unter den Datenbanktools in der oberen Menüleiste. Nachdem diese Funktion geöffnet ist, erscheint ein Auswahlfenster. Dort sind alle Tabellen auszuwählen, die in Beziehung zueinander gestellt werden sollen. Im Fall dieser Datenbank Anwendung sind das alle Tabellen, außer der Tabelle Grenzwerte.

Wie in Abbildung 24 zu erkennen ist, werden die Tabellen zunächst übersichtlich auf der Oberfläche angeordnet. Um eine Beziehung zwischen zwei Tabellen herzustellen, wird das entsprechende Attribut der einen Tabelle auf das Attribut der anderen Tabelle gezogen. Dabei ist die Reihenfolge des Vorgehens wichtig. Je nach Zugrichtung entsteht die

⁴⁴ Memo ermöglicht das Hinterlegen eines langen Textes bis 63.999 Zeichen. Unter „Text“ können maximal 255 Zeichen hinterlegt werden.

⁴⁵ Beim Datentyp Anlage können Dateien hinterlegt werden. Dieses wird beispielsweise für das Abspeichern eines PDF oder Ähnlichem benötigt.

Richtung der Beziehung, also welches Attribut von welchem abhängt. Bei Access gibt es nur zwei Beziehungsarten, die 1-1 Beziehung und die 1-n Beziehung, wobei unter Access die 1-n Beziehung für eine 1-mc Beziehung steht. Ein Tupel in der einen Tabelle kann also auf beliebig viele Tupel in der anderen Tabelle verweisen. Es handelt sich dementsprechend nur um eine andere Bezeichnung für die gleiche Funktion. In der Abbildung 24 wird die 1-n Beziehung mit den Symbolen 1 und ∞ dargestellt. Bei einer 1-1 Beziehung befinden sich keine Kennzeichnungen an den Beziehungslinien.

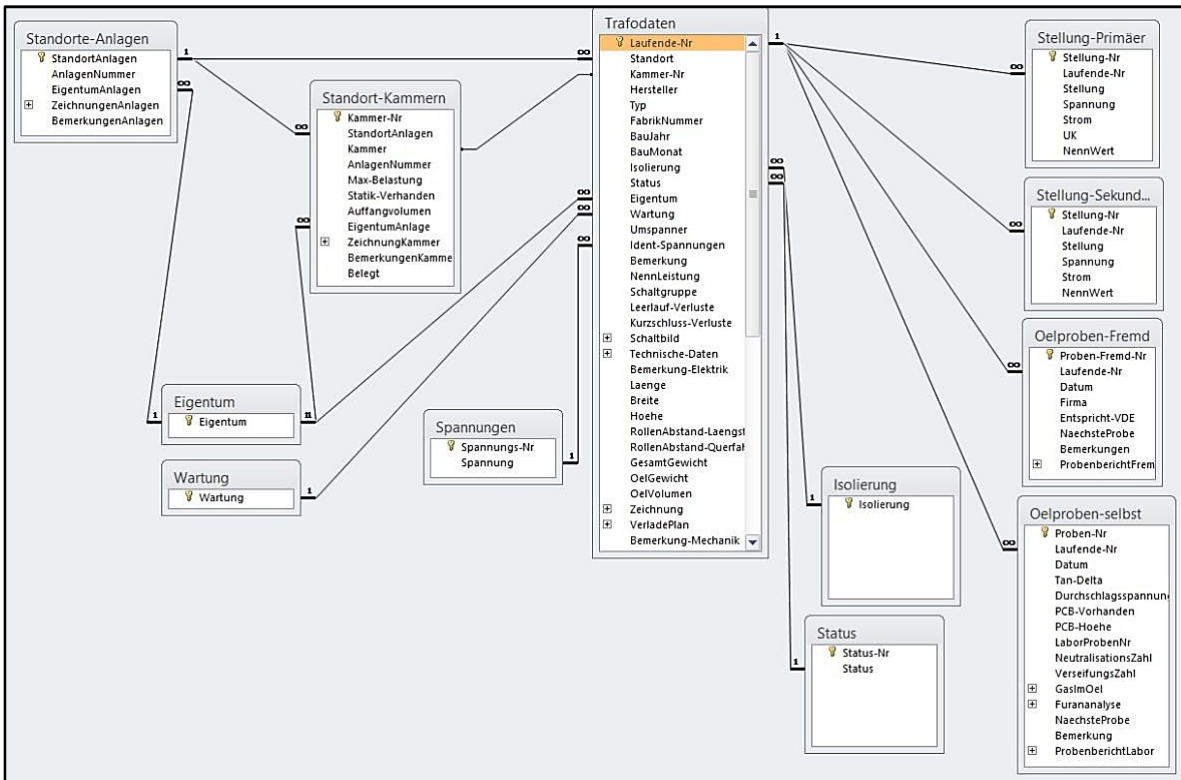


Abbildung 24: Beziehungen zwischen den Tabellen

Wenn eine Beziehung durch Drag & Drop hergestellt wird, öffnet sich nach dem Lösen des Mauszeigers ein weiteres Fenster. Dieses ist in Abbildung 25 zu sehen. Innerhalb des Fensters werden die Beziehungsarten festgelegt.

Abbildung 25: Beziehungseinstellungen unter Access 2010

Zu sehen ist die Attributauswahl der jeweiligen Tabellen, welche in Beziehung zueinander gestellt werden sollen. Dabei werden die Attribute angezeigt, die gerade via Drag & Drop aufeinander gezogen wurden. An dieser Stelle können sie mit der Auswahlliste noch geändert werden. Bei einer 1-1 Beziehung müssten die Haken für die referenzielle Integrität, Aktualisierungsweitergabe und Löschweitergabe entfernt werden. Bei der hier abgebildeten 1-n Beziehung müssen sie dagegen bestehen bleiben. Die Auswahl der referenziellen Integrität bezieht sich auf eine Aktualisierungsweitergabe an verwandte Datensätze. Es muss sich also immer ein aktueller Datensatz der einen Tabelle auf einen in der zweiten Tabelle beziehen. Die Aktualisierungsweitergabe gibt den Wert der Ursprungstabelle an die zweite Tabelle weiter. Somit können anschließend in der zweiten Tabelle Daten in Beziehung zur ursprünglichen gestellt werden. Die Löschweitergabe dient der Auflösung von allen Datensätzen, die in Beziehung zum ursprünglichen Datensatz stehen. Diese Funktion darf nur verwendet werden, um laufende Daten zu bereinigen. Bei Daten, die als Zulieferdaten für die Mastertabelle (Trafodaten) verwendet werden, darf diese Funktion nicht aktiviert sein.

Die Abbildung 24 zeigt alle Tabellenbeziehungen der aktuellen Datenbankanwendung. Zu sehen ist, dass die Tabellen „Stellung-Primär“, „Stellung-Sekundär“, „Ölproben-selbst“ und „Ölproben-Fremd“ von der Tabelle „Trafodaten“ abhängig sind. Bei diesen Beziehungen wurde auch die Löschweitergabe aktiviert. Der Hintergrund ist, dass ein Löschen des Datensatzes in der Tabelle „Trafodaten“ auch ein Löschen der Daten in direkter Beziehung bedingt. Wenn also ein Transformator gelöscht wird, interessieren auch seine Ölproben und seine Spannungs-Stellungen nicht mehr.

Genau anders herum verhält es sich bei den übrigen Beziehungen. Die Tabellen „Standorte-Anlagen“, „Standorte-Kammern“, „Eigentum“, „Wartung“, „Spannungen“, „Isolierung“ und „Status“ müssen erhalten bleiben, auch wenn ein Datensatz in der Tabelle „Trafodaten“ gelöscht wird. Daher ist auch die Beziehung genau anders herum angeordnet als bei den vorher beschriebenen Tabellen. Auch existiert bei diesen Beziehungen keine Löschweitergabe. Das rührt daher, dass die Attributwerte auch nach einem gelöschten Transformator in der Tabelle „Trafodaten“ noch weiterhin benötigt werden. Sonst stünden diese für die Auswahl bei einem neuen Datensatz nicht mehr zur Verfügung. Die Abhängigkeiten sind ersichtlich an der Anordnung der 1 und des ∞ -Zeichens an den Beziehungslinien.

Nachdem nun alle Tabellen in Beziehung zueinander gestellt sind, ist das Datenschema der Datenbankanwendung fertiggestellt. Damit ist der Kern der Datenbank, wie er in Kapitel 5.2 beschrieben ist, gemeint. Theoretisch ist an dieser Stelle schon eine komplette Verwendung der Anwendung möglich. Es müssten die Daten nur direkt in die Tabellen eingegeben werden, was nicht zu empfehlen ist.

Um die Erstellung des Datenschemas der Datenbank beenden zu können, wird an dieser Stelle noch die globale Normalisierung durchgeführt. Das bedeutet, dass die Normalformen und die Strukturregeln aus den Kapiteln 5.5 und 5.6 angewendet werden. Sollte an

Bezüglich der Datenredundanzen gibt es an dieser Stelle Auffälligkeiten. So kommen die Bezeichnungen „Standort-Anlagen“, „Laufende-Nr“ und weitere öfter in unterschiedlichen Tabellen vor. Es kann aber nicht von redundanter Datenspeicherung gesprochen werden, da die Fremdschlüssel nur der Herstellung von Beziehungen unter den Tabellen dienen. Daher müssen sie sogar mehrfach vorkommen, um in den jeweiligen Tabellen auch eine gewisse Eindeutigkeit zu haben. Eine Verknüpfung mit anderen Attributen wäre an dieser Stelle nicht zulässig, auch wenn dadurch die Redundanzen vermieden werden würden.

6.4 Benutzermaske durch Formulare erstellen

46

Es wird ein neues Formular hinzugefügt. Wie in Abbildung 26 zu sehen ist, kann auf diesem die Benutzeroberfläche erstellt werden. Über den Button „Vorhandene Felder hinzufügen“ wird eine Feldleiste geöffnet. Diese ist rechts in der Abbildung zu sehen. In dieser Liste sind alle Tabellen enthalten. Die jeweiligen Attribute dieser Tabellen stehen zur Auswahl. Auch werden an dieser Stelle die Slavetabellen angezeigt, die in Beziehung zur Mastertabelle (Trafodaten) stehen. Durch die Beziehungen zwischen den Tabellen ist es möglich, auch die Attribute der Slavetabellen im selben Formular zu nutzen.

Bei dem grundsätzlichen Aufbau der Benutzeroberfläche wird sich an einer Oberfläche orientiert, welche auf einem 19“-Bildschirm komplett sichtbar ist. Diese Maßnahme soll unnötiges Scrollen auf dem Bildschirm vermeiden. Aufgrund der Fülle an zu erfassenden Informationen wird sich für eine Aufteilung der Themengebiete in Untergruppen entschieden. Diese sind durch Registerreiter erreichbar. Durch diese Maßnahme wird eine gute Übersichtlichkeit erzeugt und der Bildschirminhalt auf der Benutzeroberfläche wirkt später nicht überfrachtet. Die einzelnen Gruppen bilden jeweils ein sogenanntes Unterformular. Zu sehen sind die einzelnen Unterformulare links in der Abbildung 26.

Um die spätere Benutzeroberfläche mit Bedienmöglichkeiten zu füllen, müssen die Attribute der Tabellen auf dem Formular platziert werden. Hinzugefügt werden Attribute, indem sie in der „Feldliste“ (rechts in der Abbildung 26) ausgewählt werden, um sie auf dem Formular per Drag & Drop zu platzieren. Nachdem diese Attribute platziert sind, lassen sich noch die Eigenschaften und die Beschriftung anpassen. Auch kann an dieser Stelle über die Feldeigenschaften entschieden werden, ob es ein Feld mit Freitexteingabe sein soll oder ein Auswahlfeld mit vorgegeben Optionen. Auf die Feldeigenschaften kann mit Hilfe des Buttons im oberen Bildbereich umgeschaltet werden. Dort finden sich auch weitere Tools zum Erstellen der Oberfläche. Im mittleren, oberen Bereich sind auch die Schaltflächen zu finden, mit denen Registersteuerelemente und Unterformulare zu erstellen sind. Registersteuerelemente werden auf diesem Formular für die Navigation zu den Unterformularen „Trafo“, „Maße“, „Ölproben“, „Standort/Kammern“, „Stammdaten“ und „Auswertung“ verwendet.

Ebenfalls ist in der Abbildung 26 ersichtlich, dass die Unterformulare für die „Stellungen“ der Primär- und Sekundärseite eigene Tabellen bilden. Diese Einstellungen lassen sich ebenfalls auf dem Eigenschaftenblatt ändern. Über den Eintrag der verschiedenen Ansichten lassen sich solche Änderungen für alle Elemente vornehmen. Durch Umschaltungen der Formularbetrachtung durch den Button oben links in der Abbildung 26 in die „Formularansicht“ kann die spätere Benutzeroberfläche betrachtet werden.

In der Benutzeroberfläche, wie sie in Abbildung 27 zu sehen ist, sind im oberen Bereich die Grundinformationen zum Transformator angeordnet. In den Registerelementen sind die vertiefenden Informationen untergebracht. Die Schaltflächen für die Pflege und Auswertung der Datenbank sind dort ebenfalls zu finden. Zu erkennen ist an dieser Stelle auch, dass die Schaltflächen „Identität“, „Isolierung“, „Status“ sowie weitere als Auswahllemente zu Verfügung stehen. Diese Funktion wurde bereits bei der Tabellensammlung berücksichtigt. Zur Auswahl stehen jeweils die Inhalte der hinterlegten

Tabellen. Die Einstellungen dafür werden bereits in der Entwurfsansicht getätigt, auch über das Eigenschaftenblatt der jeweiligen Schaltfläche. Die Ansicht, wie sie in Abbildung 27 dargestellt ist, bekommt der Benutzer nach dem Start der Anwendung zu sehen. Es stellt demnach die Startoberfläche da. Des Weiteren kann für Access eine geschützte Formularansicht (Exklusiv-Modus) aktiviert werden. Sollte dieses zum Tragen kommen, ist die Registerkarte „Stammdaten“ nicht für jeden Benutzer sichtbar.

The screenshot shows the 'Trafodaten' form with the following fields and sections:

- Header:** 'Trafodaten' tab, 'Bemerkung' field.
- Form Fields:**
 - Laufende-Nr.: 5
 - Identität: 30/6kV, HF
 - Hersteller: , Fabriknummer:
 - Typ: , Isolierung: Öl
 - Baujahr: , Status: installiert
 - Standort: Brunnenfeld
 - Kammer / Umsp: A 11 / 13
 - Eigentum: TZK - Kraftwerk
 - Wartung: TZK
- Navigation Tabs:** Trafo, Maße, Ölproben, Standort / Kammern, Stammdaten, Auswertung.
- Primary Data Table (Stufung/Spannung/Strom - primär):**

| St | Spannung | Strom | UK | Nei |
|----|----------|-------|----|-------------------------------------|
| 1 | 30000 | 40 | 3 | <input checked="" type="checkbox"/> |
| * | 0 | 0 | | <input type="checkbox"/> |
- Secondary Data Table (Stufung/Spannung/Strom - sekundär):**

| St | Spannung | Strom | Nei |
|----|----------|-------|-------------------------------------|
| 1 | 6000 | 400 | <input checked="" type="checkbox"/> |
| * | 0 | 0 | <input type="checkbox"/> |
- Other Fields:**
 - Nennleistung: kVA
 - Schaltgruppe:
 - Leerlaufverluste:
 - Kurzschlussverluste:
 - Bemerkung:
- Footer:**
 - Datensatz: 1 von 1, Kein Filter
 - Schaltbild:
 - Technische Daten:

Abbildung 27: Benutzeroberfläche des Formulars „Trafodaten“

In der Registerkarte „Trafo“ ist die Datenhinterlegung für die relevanten Spannungsdaten eines Transformators zu finden. Bei den Stellungen ist als Beispiel nur ein Eintrag zu finden. Je nach Transformator können dort aber mehrere Einträge hinterlegt werden. Über die Flächen „Schaltbild“ und „Technische-Daten“ können Dateien hinterlegt werden. Diese Funktion ist möglich, da bei der Tabellenerstellung unter den Attributeigenschaften die Option „Anlage“ gewählt wird. Somit wird beim Hereinziehen des Attributes aus der Feldliste automatisch diese Eigenschaft erkannt und umgesetzt. Beim Platzen des Mauszeigers auf dem Feld erscheinen in der Benutzeroberfläche die Auswahlmöglichkeiten. Diese sind „Datei hinzufügen“, „Datei löschen“ oder „Datei ansehen“. In den nachfolgenden Abbildungen 28 bis 31 sind die Benutzeroberflächen der Registerreiter „Maße“ bis „Stammdaten“ zu sehen.

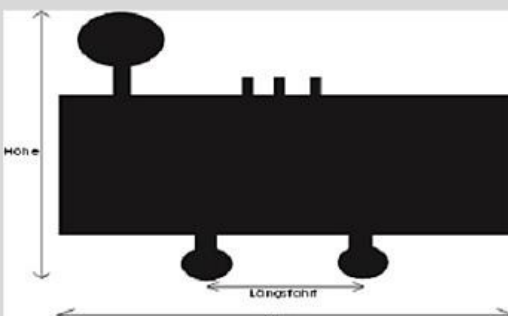
| Trafo | Maße | Ölproben | Standort / Kammern | Stammdaten | Auswertung |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|--------------------------------------------------------------------------------------------|--------------------|------------|------------|
| <div style="display: flex; justify-content: space-between;"> <div style="width: 40%;"> <p>Länge <input type="text"/> m</p> <p>Breite <input type="text"/> m</p> <p>Höhe <input type="text"/> m</p> <p>Rollenabstand <input type="text"/> m</p> <p>Längsfahrt <input type="text"/> m</p> <p>Querfahrt <input type="text"/> m</p> <p>Gesamtgewicht <input type="text"/> t</p> <p>Ölgewicht <input type="text"/> t</p> <p>Ölvolumen <input type="text"/> 0,00 m³</p> </div> <div style="width: 55%; text-align: center;">  </div> </div> | | | | | |
| <p>Bemerkung</p> <div style="border: 1px solid black; height: 60px;"></div> | | | | | |
| <p>Zeichnung</p> <div style="border: 1px solid black; width: 80px; height: 30px;"></div> | | <p>Verladeplan</p> <div style="border: 1px solid black; width: 80px; height: 30px;"></div> | | | |

Abbildung 28: Unterformular „Maße“ zur Datenbankanwendung

Die Abbildung 28 zeigt das zweite Unterformular, welches über die Registersteuerelemente zu erreichen ist. Es handelt sich um das Formular „Maße“. In ihm sind alle relevanten Abmessungen eines Transformators hinterlegbar. Die Schaltflächen stammen aber aus der Tabelle „Trafodaten“. Es handelt sich demnach um keine zusätzliche Tabelle für die Daten. Diese sind bei den anderen Hauptdaten zum jeweiligen Tupel untergebracht. Über die Schaltflächen „Zeichnung“ und „Verladeplan“ lassen sich auch an dieser Stelle Dateien hinterlegen.

| Trafo | Maße | Ölproben | Standort / Kammern | Stammdaten | Auswertung |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|----------|--------------------|------------|------------|
| <p>Labor - Ölproben</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 40%;"> <p>Datum <input type="text"/></p> <p>Proben-Nr. <input type="text"/> (Neu)</p> <p>Durchschlagsspannung <input type="text"/> 0 V</p> <p>PCB Höhe <input type="text"/> 0</p> <p>Tan-Delta <input type="text"/> 0 10³</p> <p>Neutralisationszahl <input type="text"/> 0 mg KOH/g</p> <p>Verseifungszahl <input type="text"/> 0</p> <p>Probenbericht <input type="text"/></p> </div> <div style="width: 40%;"> <p>Nächste Probe <input type="text"/></p> <p>PCB über Grenzwert <input type="checkbox"/></p> <p>Bericht Furananalyse <input type="text"/></p> </div> <div style="width: 20%;"> <p>Bemerkung</p> <div style="border: 1px solid black; height: 80px;"></div> </div> </div> <p>Bericht Gas im Öl <input type="text"/></p> <p>Datensatz: 1 von 1 Kein Filter Suchen</p> | | | | | |
| <p>Fremd - Ölprobe</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 40%;"> <p>Datum <input type="text"/></p> <p>Firma <input type="text"/></p> <p>Entspricht VDE <input type="checkbox"/> Probenbericht <input type="text"/></p> <p>Nächste Probe <input type="text"/></p> </div> <div style="width: 20%;"> <p>Bemerkungen</p> <div style="border: 1px solid black; height: 80px;"></div> </div> </div> <p>Datensatz: 1 von 1 Kein Filter Suchen</p> | | | | | |

Abbildung 29: Unterformular „Ölproben“ zur Datenbankanwendung

Bei der Registerkarte „Ölproben“, die in Abbildung 29 zu sehen ist, sind zwei Unterformulare angeordnet. Über diese werden die Daten in den separaten Tabellen „Ölproben-selbst“ und „Ölproben-fremd“ abgelegt. Die Anzahl der hinterlegten Proben ist jeweils im unteren Bereich der zwei Fenster zu sehen. Aktuell ist bei beiden Fenstern nur 1 von 1 Datensätze hinterlegt, da noch keine Daten eingetragen sind. In dem gleichen Bereich kann über die Datensatzschaltflächen auch zwischen den hinterlegten Datensätzen navigiert werden.

Abbildung 30: Unterformular „Standort/Kammern“ zur Datenbankanwendung

Abbildung 30 zeigt zwei weitere Unterformulare. Einmal das zum Anzeigen der Standorte und ein zweites zur Anzeige der Kammern. Zu sehen ist, dass die Einträge des Standortes und der Kammern mit den Einträgen aus dem Hauptformular in Abbildung 27 übereinstimmen. In diesen Unterformularen sind weitere Details zu betrachten. An dieser Stelle wird auch zwischen dem Eigentümer des Transformators und dem Eigentümer der Kammer unterschieden. Dieses können unterschiedliche Abteilungen des Unternehmens sein. Relevanz finden die Daten bei der Instandhaltung und Aufgabenverteilung bezüglich des Wasserhaushaltsgesetzes. Je nachdem welche Kammer und welcher Standort in dem Hauptformular ausgewählt sind, wird in der Abbildung 30 der passende Datensatz zu sehen sein.

Die Pflege dieser Daten erfolgt über die Schaltflächen der Abbildung 31. Über diese Buttons lassen sich jeweils die Tabellen zu den hinterlegten Daten erreichen. Es wird eine Formularoberfläche geschaffen, mit deren Hilfe die Datenänderung möglich ist. Geändert werden die Daten der Slavetabellen, welche sich hinter den entsprechenden Beschriftungen der Schaltflächen befinden. Der Zugang zu diesen Daten und der Registerschaltfläche „Stammdaten“ wird nur in einem so genannten Exklusiv-Modus möglich sein. Über ihn lassen sich bestimmte Bereiche der Anwendung schützen. Erstellt wird ein solches For-

mular auf die gleiche Weise wie das Hauptformular, über die Entwurfsansicht zu den jeweiligen Formularen.

Die Pflege von Stammdaten hat bei der Datenbankanwendung eine starke Relevanz. Da aktuell eine Sanierung der Trafokammern und Standorte nach dem Wasserhaushaltsgesetz vollzogen wird, befinden sich auch die Daten zu den Kammern und Standorten unter einer ständigen Änderung. So ist es möglich, an einer zentralen Stelle auf die Daten zuzugreifen und diese zu ändern. Das geschieht unabhängig vom gewählten Trafodatensatz. So stehen im Hauptformular immer die richtigen Daten bereit. Die Änderungen zu den Grenzwerten bezüglich der Ölproben finden im Kapitel 6.7 weitere Verwendung.

The screenshot shows a software interface with a horizontal tab bar at the top. The tabs are labeled: 'Trafo', 'Maße', 'Ölproben', 'Standort / Kammern', 'Stammdaten', and 'Auswertung'. The 'Stammdaten' tab is currently selected. Below the tabs, the main area contains eight rectangular buttons arranged in a grid-like fashion. The buttons are labeled as follows: 'Trafostatus bearbeiten' (top left), 'Eigentümer bearbeiten' (top middle), 'Anlagen / Standorte bearbeiten' (top right), 'Trafoisolierung bearbeiten' (middle left), 'Wartungseigentümer bearbeiten' (middle middle), 'Trafokammern bearbeiten' (middle right), 'Spannungsgruppen bearbeiten' (bottom left), and 'Grenzwerte für Ölproben bearbeiten' (bottom middle). The buttons are white with black text and are set against a light gray background.

Abbildung 31: Unterformular „Stammdaten“ zur Datenbankanwendung

In dem Registerreiter „Auswertung“ befinden sich weitere Buttons. Mit ihrer Hilfe ist es möglich, einen Bericht zu dem aktuell dargestellten Datensatz anzufertigen, einen Bericht zum aktuellen Sucherergebnis zu erstellen oder entsprechende Konvertierungen zu Word oder Excel vorzunehmen. Diese Funktionen werden im Kapitel 6.6 näher erläutert.

6.5 Abfrage von Datenbeständen

Eine weitere wichtige Funktion ist die Filterung und Abfrage von Datenbeständen. Dafür hat Access ebenfalls eine vorgefertigte Funktionsweise implementiert. Mit ihr ist es möglich, eine Abfrage auf bestimmte Inhalte in den Tupeln durchzuführen. Die Ergebnismenge wird standardmäßig in einer Tabelle wiedergegeben. Die Funktion wird aber etwas abgeändert, da die Filter- und Abfrageergebnisse in der Benutzeroberfläche angezeigt werden sollen.

Im ersten Schritt wird eine Abfrage für den Datenbestand erstellt. Dieses kann über die Schaltfläche „Abfrage-Entwurf“ in der Abbildung 22 des Kapitels 6.2 geschehen. Über die Schaltfläche wird ein neues Fenster geöffnet, wie es in Abbildung 32 zu sehen ist.

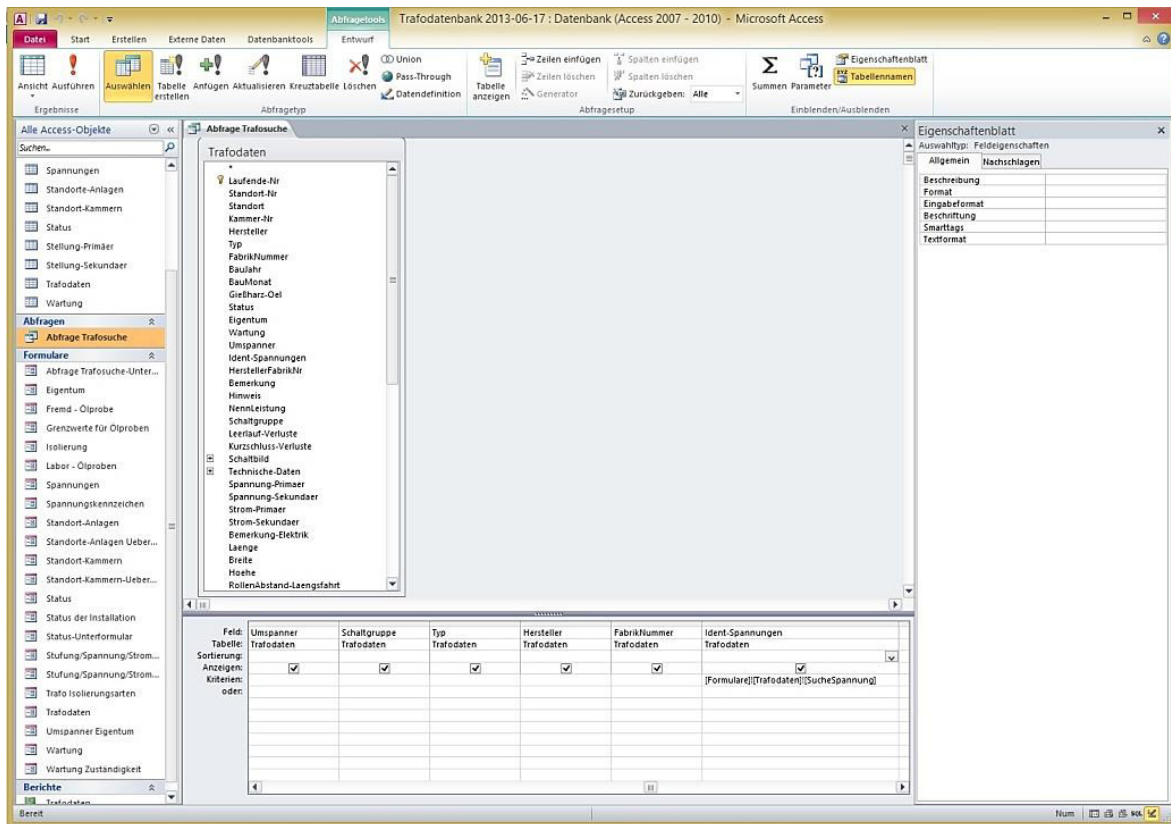


Abbildung 32: Entwurfsoberfläche für Abfragen in Access 2010

In diesem Fenster wird die Abfrage zusammengestellt. Als Erstes wird wieder die Tabelle ausgewählt, für welche die Abfrage erfolgen soll. Dieses geschieht mit einem Doppelklick in der linken Auswahlliste. In der Abbildung ist zu sehen, dass die Tabelle „Trafodaten“ für die Abfrage verwendet werden wird. Da in dieser Tabelle alle relevanten Daten gespeichert sind, ist dieses die sinnvollste Möglichkeit. Im nächsten Schritt werden die Attribute ausgesucht. Sie stellen später die Suchkriterien dar. Ausgewählt werden diese ebenfalls per Doppelklick auf das jeweilige Attribut in der Tabelle „Trafodaten“. Alle ausgewählten Attribute erscheinen in der Feldliste. Sie ist im unteren Bereich der Abbildung 32 zu sehen. Für den nächsten Schritt gibt es zwei Möglichkeiten, die Kriterien für die Suchfelder festzulegen. Zum einen könnten die Kriterien direkt in das Feld „Kriterien“ eingetragen werden. Dieses befindet sich bei den Auswahlmöglichkeiten zu den Attributen. Davon ist aber abzuraten. Jeder Benutzer müsste umständlich die Suchabfrage öffnen, um dort andere Kriterien einzutragen. Daher empfiehlt sich eine zweite Möglichkeit. Die Kriterien werden über Auswahlfenster gewählt, die sich auf der Benutzeroberfläche befinden. In der Programmierung der Datenbank erfolgt eine Verlinkung dieses Auswahlfeldes zu den Kriterien in der Suchanfrage. Dafür werden zunächst die Auswahlfelder auf dem Benutzerformular erstellt.

Um das zu ermöglichen erfolgt eine Erweiterung des Formulars „Trafodaten“ um ein paar Felder. Angeordnet werden diese im rechten Bereich der Formularoberfläche. Abbildung 33 zeigt die hinzugefügten Auswahlelemente. Für die Erstellung dieser Felder werden wieder Kombinationsfelder verwendet. Die Platzierung erfolgt via Drag & Drop. Als nächstes wird für jedes Feld die Datensatzherkunft eingetragen. Dieses ist rechts in der nachfolgenden Abbildung zu sehen. Mit dieser Datensatzherkunft wird festgelegt, welche Daten in der Listenauswahl verwendet werden können. Beispielhaft ist dieses für die Spannungsidentität dargestellt. Die Datenherkunft erfolgt aus der Tabelle „Spannungen“. Dort werden alle Attributwerte des Attributes „Spannung“ zur Verfügung gestellt, was ebenfalls rechts in der Abbildung zu sehen ist. Durch diese Auswahl ist sichergestellt, dass alle Werte, die für das Attribut „Spannung“ existieren können, auch zur Auswahl für die Suche bereitgestellt werden. Sinngemäß wird diese Prozedur auch für alle anderen Auswahlelemente wiederholt.

Abbildung 33: Erstellung der Suchkriteriums-Felder unter Access 2010

Bei „Standardwert“ für das jeweilige Listenelement wird noch „*“ eingetragen. Falls das Suchkriterium nicht verwendet wird, steht das „*“ während einer Suche für alle Möglichkeiten bei den Attributwerten. Diese Möglichkeit muss vorhanden sein, da nicht immer alle Suchkriterien genutzt werden. Es kann beispielsweise auch nur nach einem Kriterium

gesucht werden. Wichtig ist auch die Benennung der Listfelder. Anhand dieser Namen erfolgt die Zuordnung bei den Einträgen in die Suchmaske. Zu erkennen ist der aktuelle Name oben rechts im Eigenschaftsfeld der Abbildung 33. Als Name wird in diesem Beispiel „SucheSpannung“ gewählt. Für eine leichte Überprüfung der Namen empfiehlt sich ein passender und artverwandter Text. Generell kann der Name aber frei gewählt werden.

Nachdem alle Auswahlkriterien-Felder gemäß Abbildung 33 erstellt sind, müssen sie mit der Suchmaske verknüpft werden. Dieses geschieht über Einträge in die Zeile „Kriterien“ der jeweiligen Attribute in der Abfragemaske. Ersichtlich ist ein Beispieleintrag in der Abbildung 32. Dort ist für das Attribut „Ident-Spannungen“ das zugehörige Auswahlfeld bei den Kriterien eingetragen. Diese Eintragung gestaltet sich wie folgt: [Formulare]![Trafodaten]![SucheSpannung]. Die einzelnen Speicherorte sind jeweils mit „!“ voneinander getrennt und sind in eckigen Klammern angeordnet. Das verwendete Auswahlkriterien-Feld „SucheSpannung“ befindet sich auf einem Formular, im Detail handelt es sich um das Formular „Trafodaten“. So sieht es die Interpretation der oben beschriebenen Anweisung vor. Damit ist die Auswahl der Suchkriterien für die Datensatzsuche flexibel gestaltet.

Ausgelöst wird die Suche über den Button „Abfrage ausführen“, welcher in Abbildung 33 zu sehen ist. Als Funktion ist ein Makro hinterlegt. Mit Hilfe von Makros werden bei Access funktionale Abläufe gesteuert. Erstellt wird es über einen Makrogenerator, welcher nach dem Erstellen des Buttons auf dem Formular erscheint. Über ihn kann eine Funktion ausgewählt werden. In diesem Fall wird die Funktion so gewählt, dass die vorgefertigte Abfrage „Trafosuche“ mit den gewählten Kriterien ausgeführt wird. Wiedergegeben wird das Ergebnis auf dem Formular „Trafodaten“. Es sind nach der Abfrage nur noch Datensätze ersichtlich, auf welche die Suchkriterien zutreffen.

Über den Button „Alle Datensätze anzeigen“ wird das Suchergebnis zurückgesetzt und es sind wieder alle Datensätze zu sehen. Das Rücksetzen funktioniert ebenfalls über ein Makro. Dieses wird auf die gleiche Weise wie die „Abfrage ausführen“-Funktion erstellt. Beide basieren dabei auf vorgefertigten Makrobefehlen, die individuell zusammengestellt werden. Ausgewählt werden sie aus einer Liste an Funktionalitäten innerhalb des Makrogenerators. Somit ist eine komplette Funktionalität der Suchfunktion gegeben. Mit ihrer Hilfe ist es nun möglich, schnell einen oder mehrere Datensätze heraus zu filtern. Dabei handelt es sich bei endgültig über 500 Datensätzen um eine wichtige Funktion der Datenbankanwendung.

Zusammenfassend wird die Abfragefunktion anhand der Abbildung 34 beschrieben. Dabei werden über die Auswahlfelder die Kriterien ausgewählt, die für die Suche genutzt werden sollen. Diese Kriterien werden in der erstellten Abfrage verwendet. Die Abfragefunktion sucht dann nach Übereinstimmungen in den angegebenen Tabellen, in diesem Fall in der Tabelle „Trafodaten“. Sollten Kriterien mit den Attributwerten in der Tabelle übereinstimmen, wird das jeweilige Tupel, in dem das übereinstimmende Kriterium vorkommt, von der Abfrage zurückgegeben. Angezeigt werden diese Ergebnisse dann auf dem Formular „Trafodaten“.

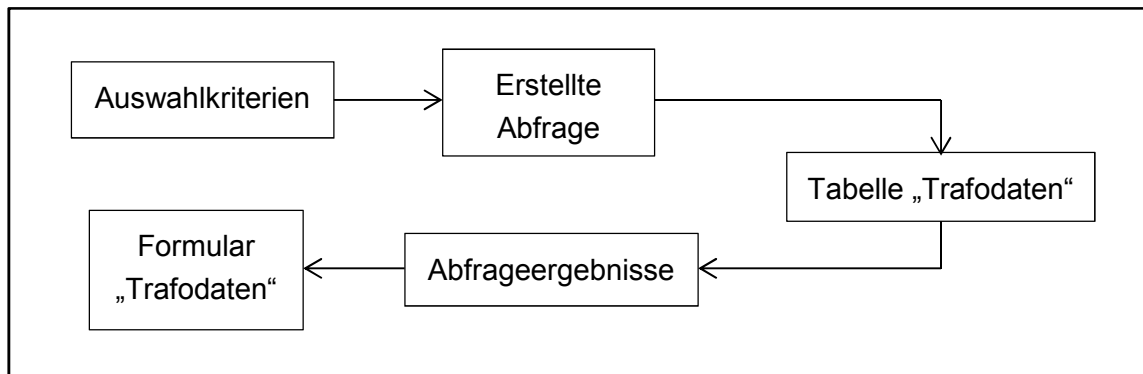


Abbildung 34: Abfragefunktion in Access 2010

6.6 Berichte aus Datenbeständen

Einen weiteren Teil des Datenbanksystems stellt der Report-Generator da. In Access wird er über die Funktion „Berichte“ umgesetzt. Ein Bericht kann individuell erstellt werden. Dazu wird die Schaltfläche „Berichtsentwurf“ genutzt, zu finden ist diese im Auswahlménü „Erstellen“, welches in der Abbildung 22 zu sehen ist.

In Erscheinung tritt eine leere Ansicht mit weißem Hintergrund. Auf dieser können wiederum Attribute verschiedener Tabellen angeordnet werden. Ausgewählt werden diese über die bereits bekannte Schaltfläche „vorhandene Felder hinzufügen“ in der oberen Menüleiste der Anwendung. Platziert werden sie nach eigenen Wünschen im Detailbereich der Entwurfsansicht. Im Berichtskopf kann eine eigene Überschrift gewählt werden, zu sehen ist sie blau hinterlegt in der Abbildung 35. Für den Seitenfuß werden die Anzeige des aktuellen Datums und die Seitenanzeige gewählt. Die Auswahl der zur Verfügung stehenden Optionen ist über die Menüleiste des Programms möglich. Es können auch individuelle Logos oder ähnliche Inhalte in den Bericht integriert werden. Wie in Abbildung 35 zu sehen ist, wird eine Auswahl an Kriterien getroffen, die im Bericht erscheinen sollen. Die nötigen Berichtsinformationen wurden per Interview-Verfahren abgestimmt. Somit ist sichergestellt, dass alle benötigten Attribute auch abgebildet werden.

Der Bericht selbst kann auf zwei Arten gefüllt werden. Es kann zum einen der aktuelle Datensatz dort abgebildet werden oder zum anderen das aktuelle Suchergebnis. In letzterem Fall wird immer ein Datensatz auf einer Seite abgedruckt. Der Bericht kann zunächst auf dem Monitor betrachtet werden, bevor ein Ausdruck getätigt wird, alternativ kann gleich ein Ausdruck erfolgen. Zu wählen sind diese Möglichkeiten über die Schaltflächen des Registers „Auswertung“, dargestellt in Abbildung 36.

| Trafodaten | | |
|----------------------|---------------------------|-----------|
| Laufende-Nr | Ident-Spannungen | |
| Hersteller | Typ | |
| Fabriknummer | Bau-Jahr | Bau-Monat |
| Standort | Kammer-Nr | |
| Umspanner | | |
| Isolierung | Status | |
| Eigentum | Wartung | |
| Nennleistung | | |
| Schaltgruppe | | |
| Leerlauf-Verluste | | |
| Kurzschluss-Verluste | | |
| Primärspannung | Primärstrom | |
| Sekundärspannung | Sekundärstrom | |
| Länge | Rollen-Abstand Längsfahrt | |
| Breite | Rollen-Abstand Querfahrt | |
| Höhe | | |
| Gesamt-Gewicht | Öl-Volumen | |
| Öl-Gewicht | | |
| Datum der Ölprobe | | |
| Durchschlagsspannung | PCB-Höhe | |
| Bemerkung | | |

Sonntag, 30. Juni 2013 Seite 1 von 1

Abbildung 35: Bericht zur Trafoauswertung in Access 2010

Über das nachfolgend abgebildete Register „Auswertung“ lassen sich die dargestellten Funktionen erreichen. Die Vorschau und Druckmöglichkeiten jeweils zum aktuellen Datensatz und dem Suchergebnis sind bereits erläutert. Funktionell sind diese Aktionen wieder mit einem Makro hinterlegt. Dort wird ein vorgefertigter Befehl ausgeführt, der den aktuellen Datensatz oder das Suchergebnis auf den erstellten Berichtsvordruck abbildet.

| Trafo | Maße | Ölproben | Standort / Kammern | Stammdaten | Auswertung |
|-----------------------------------------|------|----------|----------------------------------------------|------------|------------|
| Berichtsvorschau aktueller Datensatz | | | Berichtsvorschau aktuelles Suchergebnis | | |
| Aktuellen Datensatz ausdrucken | | | Aktuelles Suchergebnis ausdrucken | | |
| Aktuellen Datensatz zu Excel übertragen | | | Aktuelles Suchergebnis zu Excel übertragen | | |
| Aktuellen Datensatz zu Word übertragen | | | Aktuelles Suchergebnis zu Word übertragen | | |
| Tabelle für Umweltschutz drucken | | | Tabelle für Umweltschutz zu Excel übertragen | | |

Abbildung 36: Unterformular „Auswertung“ zur Datenbankanwendung

Bei den Übertragungsmöglichkeiten zu Excel und Word lässt sich ebenfalls der Makrogenerator einsetzen. Dieses wird beispielhaft in der Abbildung 37 dargestellt. Zunächst wird die Funktion ausgewählt. Dieses geschieht über den Aktionskatalog, welcher am rechten Rand der Abbildung zu sehen ist. Gewählt wird die Aktion „ExportierenMitFormatierung“. Im nächsten Schritt folgen die Einstellungen zu der gewählten Aktion.

Unter „Objektyp“ wird die Einstellung „Bericht“ ausgewählt. Wenn nur der aktuelle Datensatz ausgegeben werden soll, kann hier auch „Tabelle“ gewählt werden. Für den Objekt-namen wird der entsprechende Bericht beziehungsweise die entsprechende Tabelle herangezogen. Beim Ausgabeformat erfolgt die Einstellung „Excel“. Alternativ kann auch „Word“ ausgewählt werden, je nachdem zu welchem Programm konvertiert werden soll. Bei der Ausgabedatei sollte ein Speicherort angegeben sein, der bei den meisten Betriebssystemen gleich ist. In diesem Fall wird der Desktop gewählt. Sollte der angegeben Pfad für die Datei nicht existieren, erscheint eine Nachfrage bei Ausführung der Funktion. Weitere Einstellungen müssen an dieser Stelle nicht vorgenommen werden. Je nach gewählter Funktion erfolgt die Ausgabe des aktuellen Datensatzes oder des Suchergebnisses.

Die zwei Funktionsbuttons für die Umweltschutztabellen, welche ebenfalls in Abbildung 36 zu sehen sind, funktionieren auf gleiche Weise. Zusätzlich werden hier aber in der Vorlagendatei bei den Einstellungen im Makrogenerator noch Angaben vorgenommen. Bei dieser Tabelle kommen nur die Attribute „Standort“, „Kammer-Nr“, „Hersteller“, „Typ“, „Ident-Spannungen“, „Status“ und „Ölvolumen“ zum Einsatz. Daher wird unter Excel eine Datei erzeugt, die nur diese Spalten enthält. Wichtig ist dabei die gleiche Schreibweise der Attributbezeichnungen. Diese Excel-Datei wird dann unter Access als Vorlage angegeben und gespeichert. So werden bei zukünftigen Erzeugungen dieser Tabellen nur die

entsprechenden Daten erhoben. Ansonsten bleibt die Funktionsweise gleich, wie bei den anderen Exportfunktionen.

Die Exportfunktionen an sich lassen sich zu dem Teil des „Datenkonverters“ für „Import/Export“ beim Schalenmodell im Kapitel 5.2 zuzählen. Weitere Berichts- und Exportfunktionen sind für diese Datenbankanwendung nicht vorgesehen.

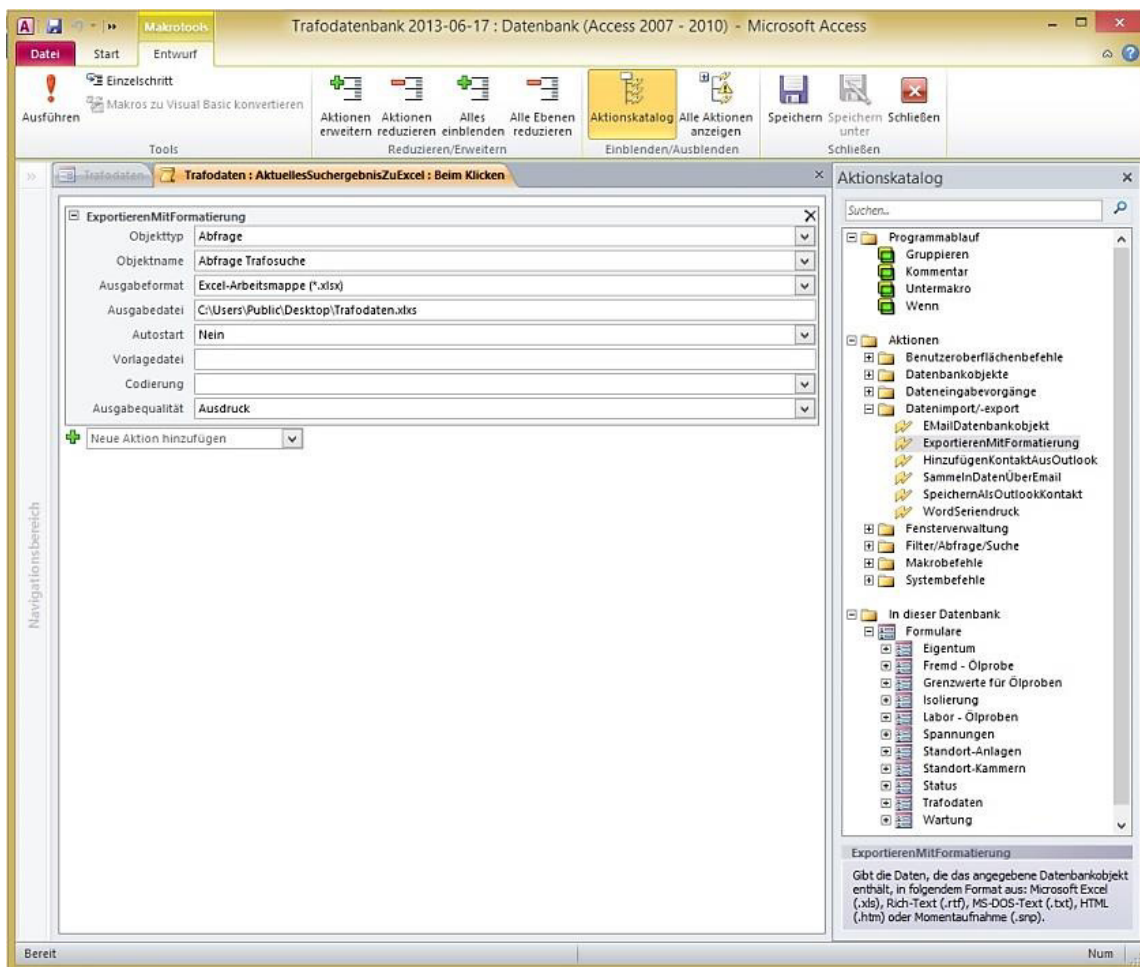


Abbildung 37: Makrogenerator für die Daten-Konvertierung zu Excel

6.7 Automatische Ölwertüberwachung

Als letzte Funktion in der Transformatoren-Verwaltungsdatenbank wird eine automatische Überwachung der Öl-Daten implementiert. Diese Funktion kann den sonstigen Programmen beim Schalenmodell zugeordnet werden. Alle dafür benötigten Programmteile sind bereits in Access integriert. Mit der so genannten „Öl-Ampel“ wird die Instandhaltung wesentlich erleichtert. Der Nutzer der Datenbank kann anhand einer optischen Signalisierung sofort einen Wartungsstau erkennen. Auf dem Hauptformular wird eine Anzeige mit einer Ampelfunktion untergebracht. Diese signalisiert bestimmte Grenzwerte zu den Öl-Daten. Dargestellt werden die Farben Weiß, Grün, Gelb und Rot. Angeordnet ist diese Anzeige im oberen Bereich des Formulars „Trafodaten“, welches das Hauptformular darstellt.

Folgende Kriterien müssen für die aufgeführten Farben erfüllt sein:

Farbe Weiß: Diese Farbe erscheint, wenn es sich um einen Gießharztrafo handelt. Die Ampel ist in diesem Fall nicht aktiv. Ein Gießharztrafo ist ein so genannter Trockentrafo, der ohne Öl als Isolationsmittel beziehungsweise Kühlmittel auskommt (siehe auch Kapitel 6.2 zur Erklärung). Da in diesem kein Öl vorhanden ist, brauchen diese Datensätze keine automatische Überwachung der Öl-Daten.

Farbe Rot: Die Überwachung der Ölwerte an sich ist nur aktiv, wenn es sich um einen Öltrafo handelt. Die Anzeige wird rot eingefärbt, wenn die gesetzten Grenzwerte nach Abbildung 38 unterschritten sind. Die Grenzwerte sind jeweils nach Spannungsebenen unterteilt, je ein Tupel steht dabei für eine Spannungsebene. Eingruppiert werden die Spannungsebenen je nach Zuordnung der Nennspannung des Transformators. Also bei einem Transformator, der eine Primärspannung von 30kV besitzt, würde der Attributwert der Nennspannung 33.000V in der Tabelle „Grenzwerte“ (Abbildung 38) betragen. Daher würde in diesem Fall das Tupel mit 33.000V für den Abgleich herangezogen werden. Eine Unterscheidung der Spannungsebenen ist wichtig, da je nach Spannungsebene unterschiedliche Entladungs- und Zersetzungsprozesse im Öl stattfinden. Es lagert sich also eine unterschiedliche Anzahl an Partikeln im Öl ab und das Öl wird unterschiedlich stark für die Isolierungserhaltung in Anspruch genommen.

| Grenzwerte | | | | | | |
|---------------|--------------|---------------------|-----------|----------------------------|------------------|------------------|
| Grenzwerte-Nr | NennSpannung | DurchschlagSpannung | Tan-Delta | DurchschlagSpannung-Gefahr | Tan-Delta-Gefahr | MaxJahreOelprobe |
| 1 | 33000 | 35000 | 500 | 20000 | 1000 | 6 |
| 2 | 120000 | 45000 | 300 | 30000 | 1000 | 6 |
| 3 | 240000 | 50000 | 200 | 35000 | 1000 | 6 |
| 4 | 1000000 | 55000 | 100 | 40000 | 1000 | 6 |
| * (Neu) | 0 | 0 | 0 | 0 | 0 | 0 |

Abbildung 38: Tabelle „Grenzwerte“ für den Öldatenabgleich

Die in der Abbildung 38 dargestellten Attributwerte werden jeweils mit den dazu passenden Attributwerten des neuesten Tupels aus der Tabelle „Ölproben-selbst“ verglichen. Sind die Schwellen jeweils unterschritten verfärbt sich die Ampel ins Rote. Herangezogen werden die Attribute „DurchschlagSpannung-Gefahr“, „Tan-Delta-Gefahr“ und „MaxJahreOelprobe“. Die übrigen Attribute werden für die gelbe Einfärbung verwendet. Des Weiteren wird für die Einfärbung der Ampel auch das Attribut „ProbeAuffaellig“ aus der Tabelle „Oelprobe-fremd“ verwendet. Sollte dieses Attribut aktiv sein, wird sich die Ampel auch rot verfärben.

Farbe Gelb: Die Funktion der gelben Farbgebung funktioniert gleich der Funktion bei der roten Einfärbung. An dieser Stelle werden aber die Attribute „DurchschlagSpannung“, „Tan-Delta“ und die Hälfte des Wertes aus „MaxJahreOelprobe“ verwendet.

Farbe Grün: Die Öl-Ampel verfärbt sich grün, wenn die Grenzwerte für gelb und rot noch nicht erreicht sind. Voraussetzung für die Funktion ist auch an dieser Stelle, dass es sich um einen Öltrafo handelt.

| Attribut | Wert |
|-------------------------------------|---------|
| Nennspannung | 33000 V |
| Durchschlagspannung Überholung | 35000 V |
| Tangens-Delta Überholung | 500 |
| Durchschlagspannung Gefahr | 20000 V |
| Tangens-Delta Gefahr | 1000 |
| Max. Jahre bis zur nächsten Ölprobe | 6 |

Abbildung 39: Unterformular „Grenzwerte für Ölproben“

Die Änderungen an den Grenzwerten können durch den Nutzer der Datenbank vorgenommen werden. Dafür muss er sich im Exklusiv-Modus der Datenbank befinden. Dieses geschieht über ein eigenes Formular, welches in Abbildung 39 zu sehen ist. Erreicht wird dieses Formular über den Button „Grenzwerte für Ölproben anpassen“, welcher in der Abbildung 31 ersichtlich ist. In dem Formular kann eine Anpassung aller Attribute vorgenommen werden, die in Abbildung 38 zu sehen sind. Über die Buttons im rechten Abbildungsbereich lassen sich die Datensätze aufrufen und bearbeiten beziehungsweise löschen oder neu hinzufügen. Nach Eingabe aller Daten wird das Formular über den entsprechenden Button wieder geschlossen. Anschließend erscheint wieder das Hauptformular. Zu allen Buttons ist in der gesamten Datenbank auch ein Erklärungstext hinterlegt. Dieser erscheint, wenn mit dem Mauszeiger über einem Button stehen geblieben wird. Beim Button „Datensatz löschen“ erscheint beispielsweise der Erklärungstext „Betätigen, um den aktuellen Datensatz zu löschen“.

Bei jedem Aufruf eines Datensatzes wird ein Abgleich durchgeführt und im Feld der Öl-Ampel auf dem Formular „Trafodaten“ präsentiert sich die aktuelle Farbe. Daher benötigt die Ampel keine Speicherung oder Ablage in einer Tabelle. Erzeugt wird das Feld für diese Funktion durch ein Textfeld, welches auf dem Formular platziert wird. Die Farbfunktionen für die Ölwertüberwachung werden mit Visual Basic in der Datenbank hinterlegt. Der Programm-Code zur Farbfunktion lässt sich im Anhang wiederfinden. Der Abgleich selbst erfolgt über einen logischen Vergleich der Attributwerte. Wie die Attribute in den Farb-Code umgesetzt werden, ist ebenfalls im Anhang zu finden.

7 Softwareintegration und Test

7.1 Integration des Datenbestandes

Nach Erstellung der Datenbankanwendung müssen in den nächsten Schritten die bestehenden Daten in die Datenbank integriert werden. Die Daten liegen sowohl in digitaler Form als auch in analoger/halb analoger Form vor. Die digitalisierten Daten befinden sich zum Teil in einer alten Access-Datenbank. Daten, die dort nicht gespeichert sind, liegen im PDF-Format oder Papierform zur Erfassung bereit. Weitere Daten müssen zunächst direkt vor Ort aufgenommen werden.

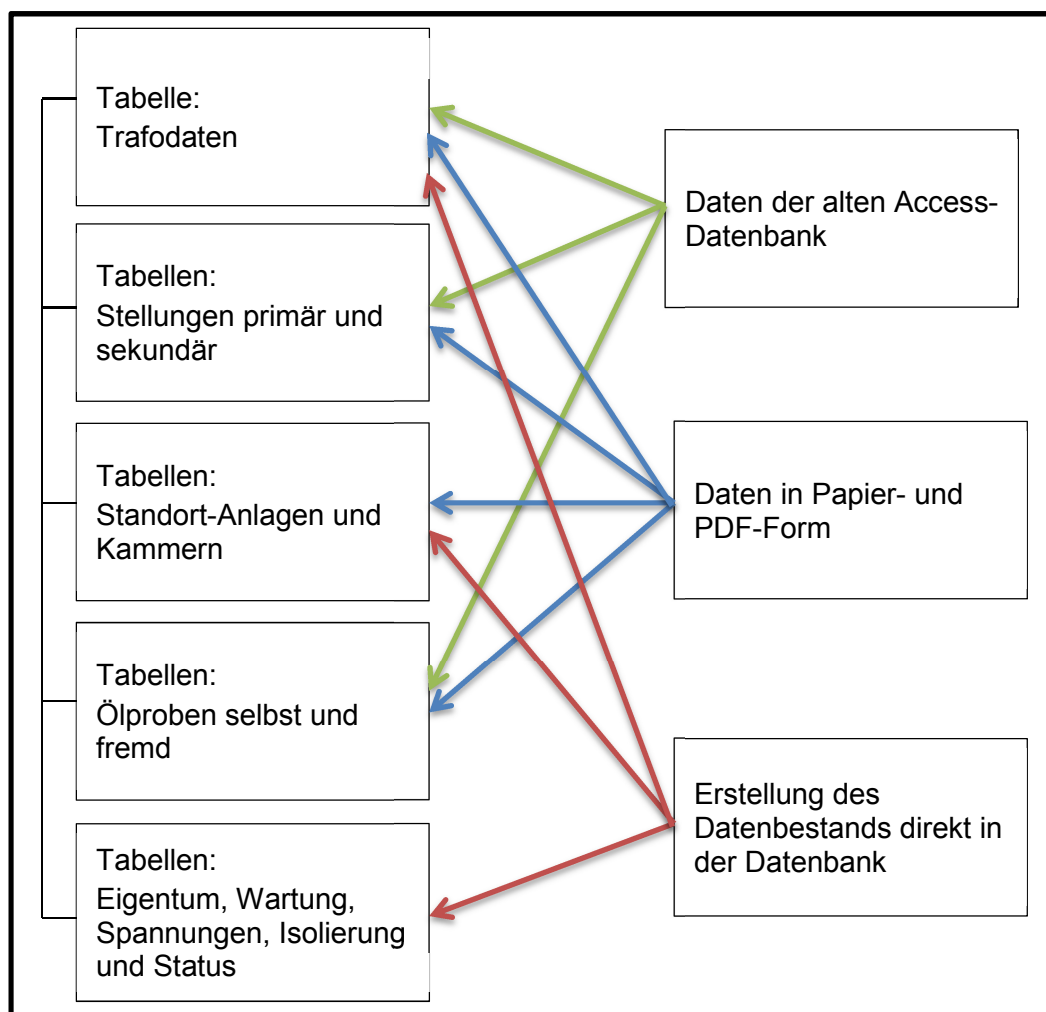


Abbildung 40: Integration der Bestandsdaten

In der Abbildung 40 sind die drei großen Blöcke bei der Datenintegration aufgezeigt. In der linken Hälfte der Abbildung befinden sich die Tabellen der neuen Datenbank, in der rechten die bestehenden Datenquellen. Insgesamt sind 549 Datensätze zu integrieren. Ein Teil der Attribute in der Tabelle Trafodaten kann aus der bestehen Access-Datenbank

übernommen werden. Dieses geschieht, indem die Tabelleninhalte angepasst und integriert werden. Dabei ist unbedingt zu beachten, dass es zu keinen Vertauschungen kommt. Dies kann zum Beispiel durch Null-Werte in den Tabellenspalten geschehen. Die vorhandenen Daten zu den Ölwerten werden aus der alten Datenbank direkt in die Tabelle „Ölproben-selbst“ übernommen. Fehlende Werte werden an dieser Stelle durch vorhandene Prüfprotokolle nachgetragen. Die Werte für die Spannungsstellungen sind in der alten Datenbank nicht komplett vorhanden, weshalb nur Teile für die Stellungen übernommen werden können. Auch an dieser Stelle werden fehlende Werte, soweit vorhanden, anhand von Prüfprotokollen zu den Transformatoren übernommen. Weitere fehlende Daten werden vor Ort anhand der Typenschilder zu den Transformatoren ermittelt. Damit sind die Transaktionen der grünen Pfeile in der Abbildung abgeschlossen.

Die blauen Pfeile stellen die Übernahme der PDFs in die Datenbank dar und die Ergänzung durch Unterlagen, die in gedruckter Form vorliegen. Die Prüfprotokolle zu den Transformatoren und Ölanalysen liegen teilweise im PDF-Format vor. Daher können diese Protokolle direkt an die neue Datenbank angehängt werden, indem sie als Anlage direkt zu den Tabellen „Ölproben-fremd“ und „Ölproben-selbst“ hinzugefügt werden. Daten zu den Standorten und Kammern werden aus vorhandenen Unterlagen entnommen und direkt in die Datenbank eingegeben. Fehlende Daten werden zeitnah ergänzt, da im Laufe der Zeit nicht mehr alle Unterlagen auffindbar sind.

Daten zu den Eigentumsverhältnissen werden aus vorhandenen Eigentumsplänen entnommen und über die Benutzermaske als neuer Datenbestand hinzugefügt. Dieses wird durch die roten Pfeile signalisiert, welche in der Abbildung 40 zu sehen sind. Gleiches geschieht auch mit den Daten zu den Wartungsbereichen. Die Attributwerte für die Tabellen „Spannungen“, „Isolierung“ und „Status“ wurden bereits bei der Erstellung der Datenbank eingegeben. Diese Werte sind fester Bestandteil des Systems und unterliegen keiner Änderung während der Lebenszeit der neuen Datenbank. Bei der Tabelle „Spannungen“ sind alle Möglichkeiten an Trafotypen (zum Beispiel 30/6kV) berücksichtigt, die es im Netzbereich des themengebenden Unternehmens gibt. Für die Isolierung sind „Öl“ und „Gießharz“ als Attributwerte hinterlegt. Beim Status gibt es „installiert“, „reserviert“ und „verschrottet“ als Wertauswahl. Damit ist die komplette Integration des vorhandenen Datenbestandes abgeschlossen.

7.2 Normalisierung der Datenbank

Als letzte Kontrolle vor der Ausgabe der Datenbank an die Anwender wird eine Normalisierung durchgeführt und die Anwendung der Strukturregeln überprüft. Da an dieser Stelle bereits alle Daten in der Datenbank erfasst sind, werden beispielsweise Redundanzen und falsche Zusammenhänge gut erkannt. Dieses fällt bereits durch die Häufung von Begriffen bei der Durchsicht der Daten auf.

Zunächst wird die erste Normalform überprüft, es müssen alle Attribute einfache Attributwerte oder Null-Werte aufweisen. Bei der Durchsicht werden keine Auffälligkeiten festge-

stellt. Dieses hätten mehrfache Attribute oder Attributkombinationen (zum Beispiel Name, Straße) sein können. Daher ist die erste Normalform erfüllt. Die zweite Normalform ist ebenfalls erfüllt, da alle Attribute voll funktional vom Identifikationsschlüssel abhängig sind. Dieses ist schon durch die Tabellenform gegeben. Jedes Tupel enthält den Identifikationsschlüssel der jeweiligen Tabelle, von daher stehen auch die Attribute in Abhängigkeit zu diesem. Transitive Abhängigkeiten sind ebenfalls nicht gegeben. Das ist auch an der Abbildung 24 im Kapitel 6.3 zu erkennen. Somit sind die ersten drei Normalformen erfüllt und der Datenbestand gilt als normalisiert.

Für die globale Normalisierung müssten die vierte und fünfte Normalform noch angewendet werden. Da aber unter Access nur in einer Domäne gearbeitet wird, erübrigt sich dieser Test. Aufgrund dessen sind globale Attribute im weitesten Sinne nicht vorhanden. Access unterscheidet nicht zwischen lokalen- und globalen-Attributen. Alle Attribute haben den gleichen Wertebereich. Aus diesen Gründen sind unter Access die beiden letzten Normalformen immer erfüllt.

An dieser Stelle muss aber die Redundanzfreiheit separat betrachtet werden. Redundanzen könnten trotz der einheitlichen Wertebereiche noch vorhanden sein. Abgesehen von der Mehrfachverwendung der Attribute, um mit ihnen die Beziehungen zwischen den Tabellen herzustellen, werden keine Redundanzen gefunden. Diese Art von Redundanz ist zulässig und im Kapitel 6 bereits erläutert worden. Damit sind alle Normalformen erfüllt und es wird von einem global normalisierten Datenbestand gesprochen.

Die Strukturregeln nach Kapitel 5.6 sind ebenfalls erfüllt. Dieses bedeutet im Einzelnen:

- Jede Tabelle besitzt einen Identifikationsschlüssel. Dieser wird beim Erstellen der Tabelle bereits vergeben und indiziert.
- Alle Tabellen befinden sich in der dritten Normalform und es existieren auch nur lokale und globale Attribute. Andere Attribute sind unter Access auch technisch nicht möglich.
- Die lokalen Attribute weisen statische Wertebereiche auf. Beim Erstellen der Tabellen wird für jedes Attribut ein fester Wertebereich eingetragen.
- Rekursive Tabellenbeziehungen sind nicht vorhanden. Dieses ist auch an der Beziehungsübersicht in der Abbildung 24 ersichtlich.
- Unter- und Obermengen-Beziehungen existieren nicht. Das wird bereits bei der Zusammenstellung von Beziehungen beachtet. Bei einer Betrachtung der Tabelleneinträge wird dieser Zustand auch noch einmal bestätigt.
- Da alle Attribute einen statischen Wertebereich aufweisen, wird dieser auch bei der Attributverwendung als Fremdschlüssel entsprechend berücksichtigt.

Die Einhaltung der Normalformen und der Strukturregeln garantiert keine komplett fehlerfreie Datenbankanwendung. Dafür müssen anschließend Funktionstests durchgeführt werden. Erst dann kann eine hohe Bugfreiheit der Software garantiert werden.

7.3 Test und Optimierung

Zum abschließenden Test der neuen Datenbankanwendung wird diese den späteren Nutzern zum Betatest zur Verfügung gestellt. Vorgesehen sind 14 Tage Testzeit. In Betracht der Funktionen handelt es sich um einen kurzen Zeitraum für einen Softwaretest. Es kann passieren, dass nicht alle Fehler gefunden werden und zu einem späteren Zeitpunkt noch einmal nachgearbeitet werden muss. Ein längerer Zeitraum steht nicht zur Verfügung, da die Datenbankanwendung in Dienst gestellt werden soll.

Für den letzten Test wird die Datenbank im Alltagsgeschehen verwendet. Die Nutzer sollen alle Funktionen ausführen und dabei auf Fehlermeldungen und Fehlverhalten der Datenbank achten. Falls Fehler auftreten, sollen diese schnellstmöglich dem Ersteller der Datenbank mitgeteilt werden.

Bei der Anwendung der Datenbank ist aufgefallen, dass in dem Auswahlbereich des Kombinationsfeldes für die Trafokammern immer alle Kammern im gesamten Werk zur Verfügung stehen. Dieses dürfte nicht passieren, da die Auswahl der Kammern durch die jeweiligen Standorte beschränkt ist. Dieser Fehler wird abgestellt, indem bei dem Datenherkunftstyp die richtige Tabelle angegeben wird. An dieser Stelle war versehentlich die Tabelle für die Abfragen hinterlegt wurden. Getauscht wird sie gegen das Attribut „Kammern“ aus der Tabelle „Standort-Trafokammern“. Nach erneuter Funktionskontrolle funktioniert die Auswahl fehlerfrei.

Ein weiterer Fehler ist bei der Suchfunktion aufgefallen. Bei Ausführung der Suche wird das Suchergebnis nicht in dem Formular „Trafodaten“ wiedergegeben, sondern in einer separaten Tabelle geöffnet. Dadurch ist das Bearbeiten der gefilterten Datensätze nur in der Tabellenansicht möglich und nicht in der Formularansicht. Fehlerursache war eine falsche Einstellung im Makro des „Abfrage ausführen“-Buttons. Dort war versehentlich die Datenblattansicht gewählt und nicht die Formularansicht. Dieser Fehler konnte ebenfalls schnell behoben werden.

Weitere Fehlfunktionen sind im Testzeitraum nicht aufgefallen. Nach den 14 Tagen wird eine stichprobenartige Überprüfung des Datenbestandes durchgeführt. Dabei werden besonders die bewegten Datensätze betrachtet. Überprüft wird, ob Fehler bei der Datenablage unterlaufen sind, also ob Daten fehlerhaft sind oder am falschen Speicherplatz abgelegt werden. Diese Überprüfung ergibt keinerlei Fehler in der Datenspeicherung. Somit sind alle Daten geprüft und die Datenbank zur Trafoverwaltung kann in Dienst gestellt werden.

7.4 Endkonzept der Datenbank

Zur Präsentation des Endkonzeptes wird in Abbildung 41 die fertige Oberfläche dargestellt. Über diese wird die komplette Datenbank bedient. Der Aufbau an sich wird bereits in Kapitel 6 beschrieben. Durch die integrierten Funktionen wird vermieden, dass der Nutzer für die Druck- und Exportfunktion tiefer in die Datenbankmaterie einsteigen muss. Die

Funktionen können übersichtlich mit einem Button ausgelöst werden. Die integrierte Öl-Ampel ist an dem grünen Balken im mittleren Abbildungsbereich zu erkennen. Im rechten Bereich der Bedienoberfläche ist die Suchfunktion untergebracht. Mit Hilfe der Auswahlkriterien können die entsprechenden Suchkriterien gewählt werden. Zur Auswahl stehen jeweils alle Möglichkeiten, die es für das betreffende Attribut gibt. Somit kann nicht nach Attributwerten gesucht werden, die nicht in der Datenbank existieren.

Abbildung 41: Benutzeroberfläche der Datenbankanwendung

Den linken Abbildungsbereich nimmt die Eingabemaske ein. Über sie werden alle Werte übersichtlich dargestellt und verwaltet. Mit Hilfe der Registerreiter kann in die weiteren Oberflächen gewechselt werden. Unter „Trafo“, „Maße“, „Ölproben“ und „Standort/Kammern“ stehen weitere Informationen zum Transformator und den Instandhaltungsdaten zur Verfügung. Unter dem Reiter „Stammdaten“ verbergen sich Buttons zur Grunddatenverwaltung. Über diese können diverse Grenzwerte oder Eigentumsverhältnisse angepasst werden, so wie es in Kapitel 6 bereits beschreiben wird. Unter „Auswertung“ stehen ebenfalls die schon erwähnten Funktionen zur Verfügung. Demnach kann die Datenbank allen gestellten Anforderungen gerecht werden.

8 Zusammenfassung

8.1 Ergebnisse

Im Rahmen dieser Diplomarbeit wurde eine umfassende Datenbank zur Verwaltung von Transformatoren erzeugt. Die hier präsentierte Lösung befindet sich auf dem Stand der aktuellen Anforderungen. Somit konnten den Nutzungsumfang betreffende Vorgaben erfüllt werden. Diese sind:

- Verwaltung von elektrischen Transformatorendaten und Abmessungen
- Verwaltung von Ölanalysen und Instandhaltungsdaten
- Verwaltung von Trafokammern und Standorten
- Verwaltung von Daten für das Wasserhaushaltsgesetz
- Informationsgewinnung für die Instandhaltung

Nachdem einleitend die Anforderungen an die Datenbank festgelegt sind, wurde im Kapitel 4 eine Marktanalyse durchgeführt. Bei dieser sind die Programme SAP NetWeaver - ABAP, Microsoft Access und eine C#-Lösung in Kombination mit einem SQL-Server in die engere Auswahl gekommen. Durch einen Kriterienvergleich konnte die passendste Programmierungsumgebung gefunden werden. Somit wurde eine gute Basis für die nächsten Kapitel gegründet.

Im fünften Kapitel wird noch einmal ein Exkurs zu den Datenbankgrundlagen vorgenommen, welcher zum besseren Verständnis der folgenden Kapitel dient. Durch die Platzierung des Kapitels kann an dieser Stelle noch einmal gezielt auf die benötigten Grundlagen verwiesen werden. Somit besteht ein guter Übergang ins sechste Kapitel.

Im Kapitel 6 wird dann die eigentliche Datenbank-Erstellung im Detail beschrieben und es erfolgt eine Übersicht über die Access-Funktionen. Diese wurden größtenteils für die Funktionalität der Datenbankanwendung benötigt. Platz für Erweiterungsmöglichkeit besteht aber auch an dieser Stelle. Durch die Anordnung aller Funktionen auf einer Oberfläche wird hier eine hohe Benutzerfreundlichkeit geboten. Abgerundet ist dieses durch eine flexible Abfragemöglichkeit und Berichtsfunktionen.

Nach der Integration des Datenbestandes wurde im siebten Kapitel ein Abgleich durch die Normalformen und die Strukturregeln durchgeführt. Ergänzt wurden diese Prüfungen durch eine Funktionskontrolle durch die späteren Anwender der Datenbank. Somit kann an der Stelle von einer annehmbaren Fehlerfreiheit gesprochen werden, auch wenn nicht viel Zeit für die Kontrollen zur Verfügung stand.

8.2 Ausblick

Das Konzept der neuen Transformatoren-Verwaltungs-Datenbank lässt noch genügend Freiraum für zukünftige Erweiterungen. Für die heutige Nutzung ist der Funktionsumfang der Datenbank angemessen. Für weitere Arbeitserleichterungen bei der Verwaltung der Transformatoren kommen aus heutiger Sicht noch folgende Möglichkeiten in Betracht:

- Suchfunktion gemäß Öl-Ampel: Die Suchfunktion könnte noch um die drei Kriterien der Öl-Ampel erweitert werden. Damit wäre es möglich, einen Wartungsstau zu erkennen, indem alle Datensätze mit roter Öl-Ampel angezeigt werden.
- Mehrbenutzerbetrieb: Einen Mehrbenutzerbetrieb zu integrieren ist möglich, da das Frontend von Access vom Backend gelöst werden könnte. Somit könnten dann mehrere Nutzer parallel an der Datenbank arbeiten, wenn diese zentral auf einem Server liegt.
- Datenauslagerung auf einen SQL-Server: Mit dieser Funktionsänderung würde eine höhere Sicherheit der Datenbank erreicht werden. Die Benutzer hätten nur noch Access als Zugriffsoberfläche und die Daten liegen zentral auf einem Server. Somit sind die Daten immer noch gesichert, auch wenn Access abstürzen würde, daher wäre in dem Fall nur die Benutzeroberfläche zerstört.

Abschließend bleibt die Erkenntnis, dass eine Software niemals komplett fertig ist. Es gibt zukünftig immer Möglichkeiten ein Programm beziehungsweise diese Datenbankanwendung noch zu verfeinern. Die Möglichkeiten sind dabei vom Einfallsreichtum des Programmierers abhängig.

Index

Datenbankmodelle

Aufbau 23

Anwendung 37

Übersicht 25

Formular

Anwendung 46

Fremdschlüssel

Anwendung 44

Funktionsweise 31

Normalisierung

Anwendung 45, 62

Funktionsweise 33

Primärschlüssel

Anwendung 43

Funktionsweise 31

Salzgitter Flachstahl GmbH

Firmenaufbau 1

Strukturregeln

Anwendung 46, 63

Funktionsweise 35

Tabellen (Entitäten)

Anwendung 38

Aufbau 27

Beziehungen 29

Tabellenbegriffe

Attribut 28

Attributwert 28

Null-Wert 28

Tupel 28

Literatur

- [A91] Appelrath, Hans-Jürgen: Datenbanksysteme in Büro, Technik und Wissenschaft – GI-Fachtagung.-1. Aufl.- Kaiserslautern: Springer, 1991, S.178-197
- [AA13] Albahari, Joseph; Albahari, Ben: C# 5.0 – kurz & gut.-3. Aufl.- Köln: O'Reilly Verlag GmbH & Co.KG, 2013, S.7-27
- [C76] Chen, Peter Pin-Shan: The Entity-Relationship Model – Toward a Unified View of Data.-1. Aufl.- Massachusetts: Institute of Technology, 1976, S.9-36
- [EGH⁺92] Engels, Gregor; Gogolla, Martin; Hohenstein, Uwe; Hülsmann, Klaus; Löhr-Richter, Perdita; Saake, Gunter; Ehrich, Hans-Dieter: Conceptual Modelling of Database Applications Using an Extended ER Model.-9. Aufl.- Amsterdam: North Holland, 1992, S.157-204
- [EN02] Elmasri, Ramez A.; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen.-3. Aufl.- München: Pearson Studium, 2002
- [EN09] Elmasri, Ramez A.; Navathe, Shamkant B.: Grundlagen von Datenbanksystemen – Bachelorausgabe.-3. Aufl.- München: Pearson Studium, 2009, S.39-130
- [G11] Geisler, Frank: Datenbanken – Grundlagen und Design.-4. Aufl.- Heidelberg: Hüthig Jehle Rehm GmbH, 2011, S.47-89
- [H10] Hölscher, Lorenz: Das Handbuch – Microsoft Access 2010.-1. Aufl.- Köln: O'Reilly Verlag GmbH & Co.KG, 2010, S.215-226
- [H11] Hölscher, Lorenz: Richtig einsteigen: Datenbanken entwickeln mit Access 2010.-1. Aufl.- Köln: O'Reilly Verlag GmbH & Co.KG, 2011, S.17-24

- [H97] Heuer, Andreas: Objektorientierte Datenbanken.-2. Aufl.- Bonn: Addison-Wesley, 1997
- [K12] Kühnel, Andreas: Visual C# 2012 - Das umfassende Handbuch.-6. Aufl.- Bonn: Galileo Computing, 2012, S.57-332
- [KE11] Kemper, Alfons; Eickler, André: Datenbanksysteme – Eine Einführung.-8. Aufl.- München: Oldenbourg Wissenschaftsverlag, 2011, S.31-66
- [KK06] Keller, Horst; Krüger, Sascha: ABAP Objects - ABAP-Programmierung mit SAP NetWeaver.-3. Aufl.- Bonn: SAP Press, 2006, S.151-190
- [KSE07] Krüger, Sascha; Seelmann-Eggebert, Jörg: ABAP – Best Practices.-2. Aufl.- Bonn: Galileo Press, 2007, S.21-345
- [M11] Minhorst, André: Access 2010 – Das Grundlagenbuch für Entwickler.-3. Aufl.- München: Addison-Wesley, 2011, S.928-933
- [P07] Preiß, Nikolai: Entwurf und Verarbeitung relationaler Datenbanken.-1. Aufl.- München: Oldenbourg Wissenschaftsverlag, 2007, S.66-82
- [P11] Piepmeyer, Lothar: Grundkurs Datenbanksysteme – Von den Konzepten bis zur Anwendungsentwicklung.-1. Aufl.- München: Carl Hanser Verlag GmbH, 2011, S.81-162
- [PH09] Poetzsch-Heffter, Arnd: Konzepte objektorientierter Programmierung - Mit einer Einführung in Java.-2. Aufl.- Berlin: Springer, 2009, S.8-21
- [S09] Steiner, René: Grundkurs relationale Datenbanken.-7. Aufl.- Wiesbaden: Vieweg & Teuber, 2009, S.5-82
- [SSH10] Saake, Gunter; Sattler, Kai-Uwe; Heuer, Andreas: Datenbanken – Konzepte und Sprachen.-4. Aufl.- Heidelber: Hüthig Jehle Rehm GmbH, 2010, S.7-157

- [SW07] Schneider, Uwe; Werner, Dieter: Taschenbuch der Informatik.-6.
Aufl.- Leipzig: Carl Hanser, 2007, S.445

Anhang

Im Folgenden ist der Programm-Code für die Einfärbung des Textfeldes in den Farben Weiß, Grün, Gelb und Rot zu finden. Dieser befindet sich als Visual Basic Code im Bereich des Formulars „Trafodaten“.

```
Private Sub Zeige_Ampel(nLaufende-Nr As Long, sIsolierung As String)
    Dim nGreen As Long, nYellow As Long, nRed As Long, nWhite As Long
    Dim AmpÖlpr As Ampel

    Me.Ampel.Value = ""
    If Not Me.NewRecord Then
        nGreen = RGB(0, 255, 0)
        nYellow = RGB(255, 255, 0)
        nRed = RGB(255, 0, 0)
        nWhite = RGB(255, 255, 255)

        If (sIsolierung = "Öl") Then
            AmpÖlpr = Get_Ampel_Ölprobe(nLaufende-Nr)

            Me.Ampel.ForeColor = nWhite
            Me.Ampel.Value = AmpÖlpr.Text

            Select Case AmpÖlpr.Farbe
            Case Grün
                Me.Ampel.BackColor = nGreen
            Case Gelb
                Me.Ampel.BackColor = nYellow
            Case Rot
                Me.Ampel.BackColor = nRed
            Case Else
                MsgBox "Farbe nicht definiert in frmTrafodaten Zeige_Ampel()", vbOKOnly
            End Select
        Else
            Me.Ampel.BackColor = Me.Detailbereich.BackColor ' kein Öltrafo
        End If
    Else
        Me.Ampel.BackColor = Me.Detailbereich.BackColor ' neuer Eintrag
    End If
End Sub
```

Ein weiterer Visual Basic Code ist für die Umsetzung des Status in eine Farbe notwendig. Mit Hilfe dieses Codes wird der festgelegte Status (Zustand der Ölproben) auf einer Farbe abgebildet. Falls ein Attributwert fehlt oder keine Ölprobe vorliegt, wird dieses mit einem Text im Ampelfenster signalisiert.

```
Private Function Get_ÖIAmpel_Text(Laufende-Nr As Long, Isolation As Variant) As String
    Dim ÖIAmpel As Ampel
    If (GetString(Isolation) = "Öl") Then
        ÖIAmpel = Get_Ampel_Ölprobe(Laufende-Nr)
        If (ÖIAmpel.Status = ok) Then
            Get_ÖIAmpel_Text = "Grün"
        ElseIf (ÖIAmpel.Status = Überholung) Then
            Get_ÖIAmpel_Text = "Gelb"
        ElseIf (ÖIAmpel.Status = Gefahr) Then
            Get_ÖIAmpel_Text = "Rot"
        ElseIf (ÖIAmpel.Status = WertFehlt) Then
            Get_ÖIAmpel_Text = ÖIAmpel.Text
        ElseIf (ÖIAmpel.Status = KeineÖlprobe) Then
            Get_ÖIAmpel_Text = "keine Ölprobe"
        End If
    End If
End Function
```


Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Diplomarbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Lahstedt, den 07. Juli 2013

Alexander Hanko